Jurassic News

Retrocomputer Magazine

Anno 3 - Numero 14 - Gennaio 2008



Jurassic News

Rivista aperiodica di Retro-computing

Coordinatore editoriale
Tullio Nicolussi [tn]

Redazione
Sonicher [sn]
redazione@jurassicnews.com

Hanno collaborato a questo numero:
Salvatore Macomer [sm]
Lorenzo 2 [L2]
Besdelsec [bs]
Maurizio Martone [mm]

Impaginazione e grafica
Anna [an]

Diffusione
marketing@jurassicnews.com

Contatti
info@jurassicnews.com

Sommario - Gennaio 2008

In Copertina

Il calcolatore Texas Instruments TI 99/4A emerge da un cumulo di macerie informatiche...
E' fra dei sistemi che hanno goduto di una certa fortuna agli esordi dell'informatica personale, grazie alla bontà del progetto e alla forza commerciale dell'azienda produttrice.

Editoriale

Si dia inizio alle danze.

Signori si parte, in carrozza per il terzo anno di Jurassic News!
Ci aspettano otto numeri della rivista da creare, cento e più articoli da studiare, scrivere, correggere, impaginare. Circa 300 fra foto e screen catturati sui video delle varie macchine, tante ore di ricerche e altrettante scoperte in quello che è il contenitore della conoscenza mondiale: Internet.
Inutile scommettere ora sulla realizzazione di quanto ci stiamo proponendo, staremo a vedere, il tempo giudicherà. Certo devo dire che mi fa piacere quando ricevo degli incoraggiamenti uniti all'incredulità per la persistenza di JN nel tempo.
Tanto che mi è venuta una riflessione su questo fatto; ci ho pensato molto e sono giunto alla conclusione che stiamo vivendo un periodo storico nel quale l'incertezza o meglio, l'inconsistenza, la fa da padrona. Difficile trovare qualcuno che si impegni veramente per un ideale o anche solo per un principio, difficile trovare chi sappia il significato delle parole "dovere" e "impegno".
Non voglio fare di tutta l'erba un fascio ma certo, soprattutto le nuove generazioni, hanno come linea guida il disimpegno.
Liberi tutti di pensare ed agire come meglio appare per loro, a noi casomai disilluderci e non fidarci delle vuote promesse, ma mi chiedo cosa vedono queste persone quando si guardano indietro? Il vuoto non li sgomenta?
Evidentemente no, forse semplicemente siamo rimasti in pochi ad avere il coraggio di girarci verso il passato per vedere come eravamo e quanta strada abbiamo fatto, con fatica, sacrificio, delusioni e scoramenti.
Ma anche con la consapevolezza della nostra forza (vorrei quasi quasi chiamarla "fede", ma non mi sembra il caso...)

[Tn]

Jurassic News

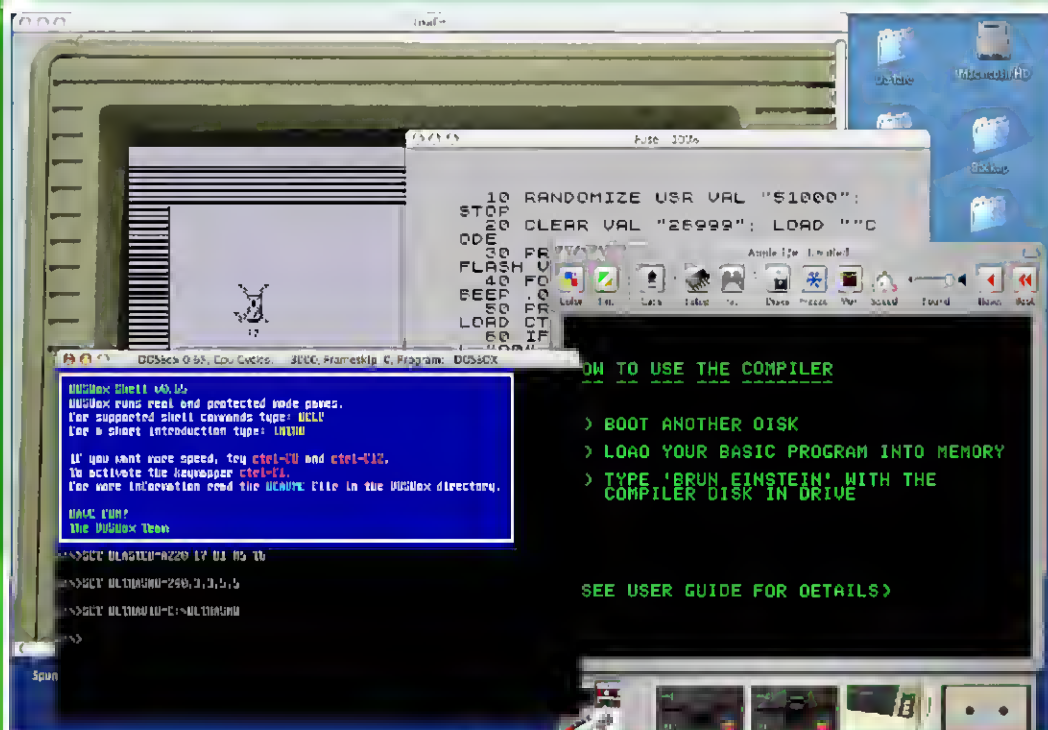
è una fanzine dedicata al retro-computing nella più ampia accezione del termine. Gli articoli trattano in generale dell'informatica a partire dai primi anni '80 e si spingono fino ...all'altro ieri.

La pubblicazione ha carattere puramente amatoriale e didattico, tutte le informazioni sono tratte da materiale originale dell'epoca o raccolte (e attentamente vagliate) da Internet.

Retrocomputing

L'àncora di salvezza

Le idee legate all'emulazione fano ormai parte della nostra storia passata, presente e futura...



Ecco sullo schermo del mio PowerBook prendere vita contemporaneamente il Lisa, un Apple IIe, uno Spectrum e il DOS.

Torniamo a parlare i questo spazio, che nella nostra idea raccoglie tutti gli aspetti culturali legati alla pratica del retro computing, di emulatori.

Non ci vogliamo dilungare in definizioni e spiegazioni su cosa essi siano o su come vengono realizzati, ne parliamo per inquadrarli nell'ambito del vasto mondo legato alla conservazione dei sistemi di calcolo.

Le idee che vogliamo sviluppare seguono i seguenti filoni principali:

- perché è importante sviluppare e mantenere gli emulatori;

- chi dovrebbe essere interessato ad essi;

- quale futuro è possibile.

Il perché esistano gli emulatori è presto detto: per simulare sistemi su altri sistemi. Sì, ma quali e quanti sistemi?

Inizialmente la gente si è messa a sviluppare questo tipo di software per usare gli applicativi di un'altra piattaforma sulla propria. In generale per disporre di maggiore scelta di software rispetto a quella nativa del sistema host. Nascono così tentativi che fanno sorridere come l'emulazione dello Spectrum su un Commodore 64 e viceversa. Più seri, anche perché con maggiore energia a disposizione, l'emulazione dell'Apple II sul Mac o del C64 su

Amiga. Per non parlare dell'emulazione del PC IBM, subito balzata alla cronaca non appena la gente capì che quella era una macchina da temere come la peste!

I tentativi di emulazione con qualche velleità professionale sono stati presto abbandonati in favore di vere schede hardware da montare sui vari Mac, Amiga e Atari ST. Questo non sorprende se pensiamo che una emulazione decente del PC è possibile oggi disponendo di macchine a 32 bit con CPU a 3 GHz e rotti, mentre fino a pochi anni orsono l'emulazione di un PC nel MAC o anche del PC nel PC era proprio l'ultima spiaggia per chi non poteva farne a meno.

Su questo fronte "produttivo", il futuro è ben delineato: la virtualizzazione. I sistemi virtuali sono ormai entrati nella logica della produzione dei data center e non se ne andranno tanto facilmente, anzi probabilmente non se ne andranno per nulla! Abituiamoci ad un futuro nel quale il nostro PC non ha installato un qualche Windows sulla macchina reale, ma ci sarà uno strato software "embeded" fra piastra madre e CPU e quello che faremo girare sarà una delle tante possibili virtualizzazioni del sistema operativo.

Probabilmente i primi ad avere bisogno degli emulatori sono stati i programmatori. Chi sviluppa per professione si trova a dover affrontare il problema della compatibilità del proprio codice con una pletora di macchine con versioni di firmware, sistema operativo e periferiche diverse. Un incubo per chi ha la necessità di allargare la propria base di installato e non può permettersi di vendere il software "chiavi in mano" con hardware e tutto il resto.

Immaginiamo che anche all'epoca d'oro dei videogiochi, che uscivano per 10-15 piattaforme diverse, il trasporto sia stato un elemento di attenzione e costo. Quindi la possibilità di vedere il comportamento di un programma su piattaforme diverse senza complicati sistemi di travaso hardware, deve essere stata la molla che ha spinto molti a dedicarsi a questo settore difficile della programmazione.

L'emulazione è uno dei settori, a mio avviso, più ostici da affrontare, non fosse altro che per la necessità di conciliare il funzionamento di due sistemi diversi e per la necessità di conoscere in maniera intima le componenti hardware della macchina che si intende emulare.

Le aziende che producono computer non dovrebbero essere contrarie all'emulazione? In fondo per esse vuol dire vendere un sistema di meno, se chi ne compra uno poi va ad emulare gli altri sullo stesso hardware! In effetti è così, ma l'opposizione non è stata né così forte né così convinta come in altre occasioni. Probabilmente si considerava il settore come una nicchia di mercato e le scarse prestazioni dei sistemi emulati non poteva impensierire. Le posizioni possono anche essere diverse e rispondere a

delle precise strategie aziendali. Ad esempio la Apple non ha mai fatto nulla per incoraggiare l'emulazione del Mac sui PC. Infatti la Apple, nonostante le ultime scelte aziendali, vende computer e se qualcuno riesce ad emulare il suo software rischia di smettere del tutto la vendita delle sue pur belle macchine. Invece essa vede di buon occhio l'esecuzione di Windows sui Mac Intel: va nella direzione originaria dell'idea di emulazione: rendere disponibili più software per la propria piattaforma.

L'emulazione del Mac o, parimenti l'esecuzione di Mac OS X su PC, non viene vista di buon occhio da nessuno (sembrerebbe!). Certo per Microsoft sarebbe riconoscere una sconfitta e non tanto per le funzionalità e l'usabilità del sistema operativo di Apple, ma per quello che ci sta sotto: Unix.

Rispetto alle idee originali alla base della realizzazione dei programmi di emulazione, ora le cose sono cambiate di brutto: ora abbiamo emulazioni della Playstation 2 su un PC, che poco hanno da invidiare al progetto originale di Sony.

Qui stiamo toccando un punto alquanto delicato che coinvolge anche aspetti legali. Infatti non si può certo sostenere che uno ha "necessità" di emulare la PS2 per lavoro! Quello che è vero invece è che oltre al risparmio dell'acquisto di una console reale, c'é anche la piaga della duplicazione abusiva del software.

Dal punto di vista tecnico e per quanto andremo a specificare nel seguito, anche questi progetti di emulazione devono essere salvati, incoraggiati e supportati!

Essi infatti fanno oggi un lavoro che sarebbe indispensabile comunque realizzare domani. Ci riferiamo alla conservazione della cultura informatica nella più ampia accezione del termine.

Troppo spesso abbiamo corso il rischio di veder sparire consistenti porzioni della cultura informatica, per rischiarne ulteriori settori solo per il rispetto di diritti di difesa del profitto immediato delle case costruttrici! E' vero che chi progetta, costruisce e commercializza un prodotto ha il diritto di ricavarne un giusto profitto, ma ci devono essere dei limiti che purtroppo oggi non esistono. E' su quel termine "giusto" che si va a discutere!

Il problema non è tanto della macchina fisica, ma del software. Sarebbe come se chi produce carta potesse evitare che questa sia usata per tramandare i classici della letteratura. Decisamente assurdo! Così chi produce una macchina di calcolo o da gioco oggi non ha il diritto di imporre che al suo abbandono, cioè quando non ha più interesse a costruirla e venderla, anche tutto il software prodotto per essa sia destinato all'oblio.

Ecco quindi la parte migliore del lavoro di un emulatore: la conservazione di quanto è stato prodotto per quel certo sistema, del quale

non esiste copia funzionante, per le future generazioni. Non si cada nella facile alzata di spalle secondo la quale "chi se ne importa: Excel XP è superato da Excel 2003 e 2007, quindi perché sprecare spazio ed energie per metterne da parte una copia funzionante?". Questo equivoco è abbastanza comune e fa parte della nostra cultura occidentale basata su un principio economico principale: consumare.

Siamo talmente abituati a consumare, anche la cultura, che giudichiamo una cosa vista come vecchia solo cinque minuti dopo averla comprata.

Una amica mi ha fatto notare, poco tempo fa, che la saga di Star Wars è "cosa vecchia". D'accordo che ha venticinque anni, bambina, ma se vuoi capire dove stai andando è meglio sapere da dove siamo venuti, non ti pare?

Molto significativo, secondo il mio parere, è il notare come nazioni meno fortunate di noi abbiano fatto tesoro delle tecnologie obsolete e quindi a buon prezzo, per sviluppare sistemi decisamente interessanti e per nulla superati. E' il caso della Russia, ad esempio, dove è attiva una industria, anzi più di una, le quali costruiscono cloni avanzati dello Spectrum Sinclair (oggi Amstrad).

Come fanno notare i fautori dell'Open Source, le tecnologie proprietarie minano la capacità di una nazione di produrre cervelli che sappiano sviluppare nuove idee. Come dire che oggi usiamo tutti il classico PC a scapito del futuro dei nostri figli!

L'emulazione è una strada possibile, a volte l'unica, che ci consente di tramandare una cultura, per cui la conclusione non può che essere una: lunga vita agli emulatori!

[Tn]

Le prove di Jurassic News

Texas Instruments TI 99/4A

Ecco un sistema di calcolo personale sul quale c'è un unanime consenso fra gli addetti ai lavori: si tratta del primo personal italiano prodotto da una grande azienda e destinato ad essere venduto in migliaia di esemplari.

Introduzione

Nel giugno 1982 giungeva in Italia un prodotto della Texas Instruments che molti stavano aspettando. Si trattava dell' home computer TI-99/4A, derivato da un prodotto precedente (TI-99/4) ma mai importato in Italia in maniera ufficiale.



La data di rilascio ufficiale negli States risale ad un anno prima ma questo ritardo si trasforma convenientemente nella maggiore disponibilità di software e di periferiche, oltre che di eventuali correzioni di bug insiti nelle ROM.

La stessa azienda lo propone sotto la dicitura "Home Computer" per distinguere la serie dalla propria linea di calcolatori scientifici. Infatti il TI-99/4 è in tutto e per tutto un "personal": nella dotazione hardware e nel software disponibile.

Sulla carta si presenta molto bene con un processore a 16 bit che promette faville in un mondo dove gli 8 bit sono uno standard. Che le promesse siano state mantenute o meno è una questione di punti di vista, certo il TI-99/4A è stato ben presto superato da altre soluzioni, forse meno dotate in termini di potenzialità, ma con l'arma del basso prezzo a loro favore.

La pubblicità che troviamo sulle riviste dell'epoca parla di una macchina capace di accettare cinque linguaggi: quattro di programmazione e come quinto l'inglese. Infatti la grande novità è la presenza di un sintetizzatore capace di riprodurre e comprendere comandi vocali. Una macchina ideale quin-

di per lo studio prima di tutto e poi per il gioco, campo dove il TI-99/4A intende calare le proprie carte, forte del processore e della dotazione hardware a corredo.

Breve storia del prodotto

Le origini si possono far risalire al 1974, quando la TI introduce la propria soluzione di un processore a 4 bit progettato per applicazioni generali (general purpose). Questo chip è siglato TMS1000 e verrà utilizzato nell'industria dei controlli industriali e dell'automazione.

L'innovazione consiste nel fatto che l'adozone di un circuito specializzato in singolo package, permette un risparmio notevole nella progettazione e costruzione dei controlli elettronici.

Appena un anno più tardi (1975) la Texas Instruments è già pronta a rilasciare il TMS9900, il primo microprocessore a 16 bit della storia dell'elettronica digitale.

Attorno a questo chip la TI costruisce la sua gamma di minicomputer, conosciuti per le sigle che giocano attorno al numero "99" (990, 9900, 9995, etc...).

Altra tappa fondamentale è la costruzione nel 1978 del primo chip di sintesi vocale, un oggetto sul quale la TI investe molto ottenendo dei buoni risultati. Dal punto di vista dei prodotti esce "Speak & Spell", una specie di calcolatrice vocale progettata per aiutare nell'apprendimento della corretta scrittura e pronuncia inglese (gli stessi inglesi si rendono conto di come sia difficile scrivere in un modo e pronunciare in un'altro...).

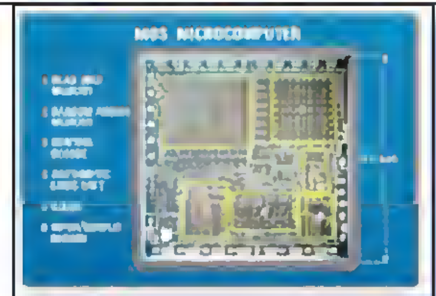
Il successo comemrciale di Speak&Spell ha generato una pletora di oggetti simili: Speak & Math™, Speak & Read™, Speak & Music™, ...

Alla fine degli anni '70 la TI pensò di essere pronta ad entrare nel mondo dei calcolatori personali e progettò il 99/4, attorno al proprio chip TMS9900 a 16 bit. La qualitù del prodotto era indubbiamente alta ma due fattori ne frenarono la diffusione: la scarsezza di software in dotazione e il costo elevato (più di 1100 dollari).

Il secondo tentativo venne progettato con il 99/4A, una versione rivista e migliorata con buona dotazione di software e con prezzo molto ribassato (attorno ai 600 dollari).

Il prodotto andò bene sopratuttto nel settore educational fino al 1982 quando il settore dell'home computer registrò la ormai famosa guerra dei prezzi che obbligò la Texsas a svendere gli stock di magazzino a meno di 100 dollari, decidendo poi di uscire definitivamente da questo settore di mercato.

Queso giochetto



Il microprocessore TMS1000 che è considerato il primo microcomputer on-chip.



Lo Speak&Spell, poco più che un gioco da bambini, ma il primo esempio di uso commerciale della sintesi vocale.

costò alla TI qualcosa come 680 milioni di dollari di perdite! Ovvio che l'azienda non ci riprovasse più, concentrandosi sul proprio core-business: l'elettronica e la strumentazione.

Primo approccio

L'unità centrale si presenta contenuta nelle dimensioni bicolore (alluminio e nero), dove trovano posto la tastiera un po' più piccola di una normale Qwerty e l'accesso ad uno slot di espansione frontale, protetto da uno sportellino a molla, adatto a contenere un modulo di espansione su ROM.

Esiste anche una versione completamente beige, uscita successivamente al lancio della prima serie.

L'alimentatore è esterno, ospitato da uno scatolotto di plastica nera che presenta una sagomatura simile all'unità centrale, con una inclinazione frontale, che assomiglia in qualche modo allo stesso componente nel Commodore 64 (ma quello del 99/4A è leggermente più piccolo).

L'accensione si effettua tramite uno switch accessibile sul frontalino, dotato di spia di accensione (due lussi per l'epoca!).

Quello che è meno indovinata è l'ingegnerizzazione della configurazione delle periferiche che sono collegate "in cascata" sfruttando uno slot presente sul fianco destro dell'unità centrale nel quale si innesta il connettore a pettine che ogni espansione porta sul suo lato sinistro. Il risultato è una catena di periferiche solidali anche meccanicamente, che diventa ingestibile se se ne possiedono più di due o tre. Se si tiene conto inoltre che l'unità floppy disk abbisogna del controller, anch'esso "scatolotto" inserito nella "catena", si rischia presto di sbordare il piano della scrivania.

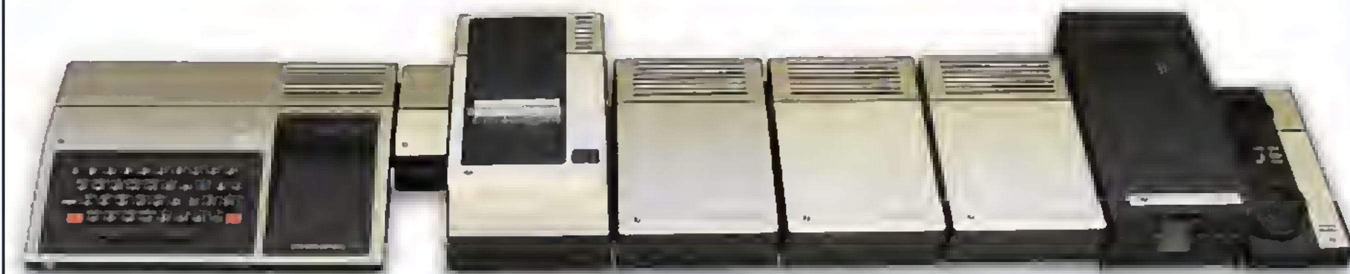
Questa configurazione sarà presto rimpiazzata in un più pratico box di espansione.

Come terminale video si può usare il classico televisore a colori sintonizzato sul canale di trasmissione del segnale da parte del computer. Altre periferiche sono collegabili tramite l'interfaccia seriale RS-232, ospitata su un modulo opzionale da collegare nella solita "catena".

Le connessioni che troviamo sull'unità centrale sono: una porta Joystick sul lato sinistro, il connettore di espansione sul lato destro e sul retro il connettore di alimentazione (proprietario), il mini-DIN per l'uscita video composita e ancora

Il TI 99/4A con la dotazione di base e l'adattatore PAL per l'Europa, visto che il progetto originale esce con quello per lo standard TNSC.





La famosa "catena" che rischia di sbordare oltre il piano della scrivania e che comunque si rivela tutt'altro che pratica. L'unico vantaggio è la titale asseza di cavi di collegamento fra una periferica e l'altra.

un 9-pin femmina con i segnali per il registratore a cassette.

L'impressione generale è di trovarsi difronte ad un buon prodotto, impressione rafforzata dal buon feedback della tastiera, dalla stabilità dell'oggetto sul piano di lavoro, dallo scarso ingombro dell'unità centrale e dalla sobrietà e serietà della colorazione scelta.

Hardware

Come non iniziare con la vera novità, cioè con il processore? Per la prima volta un 16 bit si affaccia sul mercato dei personal casalinghi e l'attesa e le speranze sono ovviamente elevate.

Il chip è il processore di produzione Texas siglato TM9900 con clock a 3 MHz. La RAM in dotazione è di appena 256 Byte di RAM ma altri 16 Kb sono disponibili come "Video RAM". L'idea è che quando si utilizza un software su cartridge presumibilmente il codice sarà tutto nella ROM del modulo di espansione, mentre la RAM video potrà servire per la grafica ad esempio nei giochi, viceversa quando si programma in Basic si potrà rinunciare a qualche features grafica per dedicare parte della RAM come memoria di programma. Probabilmente non è che si credesse molto nelle capacità di un utente casalingo di produrre in proprio programmi di grande qualità.

Il disegno del chip TMS9900 meriterebbe un articolo a parte, viste le sue caratteristiche che in qualche modo si discostano da un classico processore ad 8 bit dell'epoca. Diciamo brevemente che nasce con il TMS9900 l'idea della cache, come memoria di lavoro ad accesso immediato.

La scheda di espansione RAM esistente in due versioni porta a bordo 32 Kb di memoria (la cifra da sborsare è abnorme) per trasformare il 99 in un vero personal computer. Per usarla è però necessario

Porte di I/O: due seriali RS232 acquistando il modulo di espansione opzionale, mentre è del tutto assente la possibilità di dotare la macchina di una porta parallela per

Il chip TMS9900 in contenitore ceramico (from Wikipedia).





Il registratore ufficiale dalla texas Instruments: la periferica è siglata "Program recorder PHP 2700".

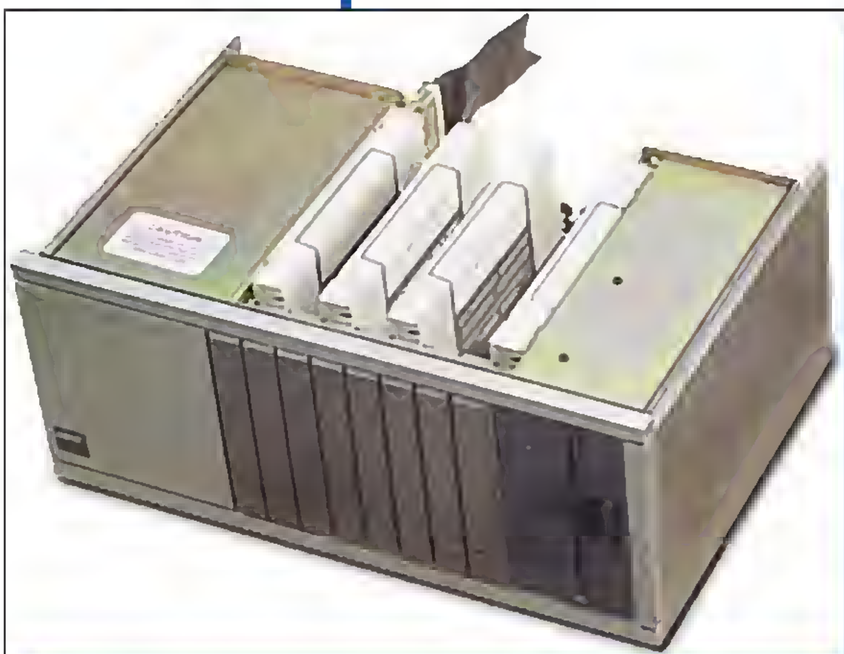
stampante.

L'alimentatore è esterno e ovviamente localizzato in relazione al paese di commercializzazione.

Il connettore video prevede la possibilità di collegare un monitor a colori della stessa TI o, via modulatore appropriato, un TV casalingo.

La memoria di massa primaria è il registratore a cassette, adeguato alla vocazione home del sistema, anche se l'espansione che permette di pilotare uno o più floppy disk dà una marcia in più alle potenzialità elaborative di questo sistema.

Il box di espansione con alcune cartdridge inserite.



I registratori a cassette utilizzabili sono due, individuati nei comandi (ad esempio LOAD e SAVE) con il parametro CS1 o CS2.

La Texas Instruments vende ovviamente una periferica certificata chiamata "Program Recorder PHP 2700" perfettamente in grado di adeguarsi alla modalità di utilizzo della macchina che è, seppur parzialmente, gestibile da comandi BASIC.

Espansioni

La possibilità di espandere il proprio sistema, aldilà delle periferiche direttamente collegabili alla "catena", passa attraverso l'acquisto del così chiamato Peripheral Expansion Box (PE Box o PEB).

Questo consiste in un box abbastanza ingombrante per la verità, in grado di ospitare un certo numero di espansioni. Come minimo, per disporre di una macchina con una marcia in più, sarebbero necessarie almeno le porte parallele e seriali, una espansione di memoria da 32K e la coppia controller-floppy disk.

Altre periferiche disponibili sono un hard disk SCSI, una scheda clock, una espansione per il video a 80 colonne. In effetti le possibilità sono ampie e permettono di far assumere alla macchina una configurazione ottimale per ogni tipo di destinazione d'uso. Non dovrebbe mancare infine il modulo "Extended Basic", obbligatorio per la gestione dei floppy e da non dimenticare (la Texas Instruments ci

tiene tanto!) il modulo di sintesi vocale.

La massima espansione del sistema (e una configurazione meccanica migliore) si ottiene acquistando il cosidetto "Box di espansione" (Peripheral Expansion Box), costituito da un robusto contenitore dove trovano posto un certo numero di cartdridge e le periferiche come il floppy.

Da punto di vista funzionale il PEB è una soluzione discreta ma realizzata con poco accorgimento al mantenimento del design; bruttissimo ad esmepio il cavo flat di collegamento con l'unità centrale.

Una variante abbastanza nota e diffusa porta il nome di interfaccia Geneve e che è a tutti gli effetti una nuova macchina.

Prodotta da Myarc nel 1987 e conosciuta anche come Myarc 9640, porta a bordo una CPU TMS9995 a 12 Mhz e può accogliere fino a 2 Mb di RAM.

Contiene un sintetizzatore musicale, l'orologio di sistema e esce su monitor con una interfaccia di tipo RGB compatibile. Dispone inoltre di un proprio sistema operativo per la gestione dei dischi, chiamato MDOS e si può comandare con una tastiera del PC IBM.

A questo punto si può mettere da parte la gloriosa unità centrale e tastiera del TI 99 e disporre di un computer completamente nuovo, come si diceva, recuperando le schede di espansione e le periferiche acquistate per la vecchia macchina.



Unità centrale e box di espansione.

Il discorso sulle espansioni disponibili meriterebbe ben più spazio di quello che possiamo dedicare a questa presentazione per cui ci fermiamo a questo punto.

Uso

All'accensione il sistema controlla l'eventuale presenza di software "bootabile" nella cassetta inserita nello slot di espansione ed eventualmente ne chiede l'attivazione in alternativa al Basic contenuto

La scheda Geneve che di fatto trasforma il TI99/4A in un sistema più moderno e potente.



```
220 OPEN #1:"CS1",INTERNAL,FIXED,OUTPUT
230 FOR X = 1 TO 5
240 PRINT #1:A$(X),B$(X),A(X)
250 NEXT X
260 CLOSE #1
270 GOTO 100
280 CALL CLEAR
290 OPEN #1:"CS1",FIXED,INTERNAL,INPUT
300 FOR X = 1 TO 5
310 INPUT #1:A$(X),B$(X),A(X)
```

Un esempio di programma Basic che interagisce con il registratore a cassettem usato come meoria di massa per la gestione di un archivio dati.

nella ROM.

Le particolarità che dispinguono questo sistema dagli analoghi e contemporanei home si devono cercare in alcune istruzioni del Basic in dotazione e nell'utilizzo delle periferiche.

Ad esempio per il registratore a cassetta sono predisposti dei comandi Basic che prevedono come parametro l'indicazione della periferica destinataria del comando: Load e SAVE sono seguiti dalle sigle CS1 o CS2 (primo e secondo registratore).

Le operazioni sono guidate da un dialogo a video che invita a fare le operazioni necessarie per posizionare il media e pigiare i giusti tasti sul registratore. Appare addirittura eccessiva questa maniacale guida per operazioni così banali e che dovrebbero essere alla portata di qualsiasi utente domestico che usi il registratore a cassette per la propria musica.

L'uso del registratore da programma si effettua con istruzioni BASIC classiche: OPEN per l'apertura del canale di comunicazione, istruzione che prevede anche qui l'indicazione della periferica coinvolta; PRINT per scrivere e INPUT per leggere dal canale aperto.

L'uso del floppy disk (una unità soft sectored da 5,25 pollici) è siglata PHP1850 e si connette al controller (esterno alla base) con un flat cable "passante", nel senso che più di un floppy può essere collegato allo stesso controller per avere "in linea" le unità individuate dalla sigla progressiva DSK1, DSK2, DSK3 (il numero massimo di unità per ogni controller è appunto tre).

Il costo di queste unità ci porta ad escludere che qualcuno potesse permettersi configurazione con più di due drive; già possederne uno sarebbe stato un lusso nel 1982!

Le stesse operazioni necessarie ad abilitare i drive successivi al primo, che prevedono l'apertura dell'involucro metallico e la rimozione del resistore di terminazione, indicano che nemmeno la Texas credeva in una vendita massiccia di configurazioni multi-drive.

Per l'utilizzo della periferica sono disponibili i comandi BASIC di comunicazione da e per i file registrati sul disco e un modulo aggiuntivo da inserire nello slot di espansione, denominato "Disk Manager", che porta a bordo i comandi di gestione dell'unità come la formattazione, il recupero del catalogo e in generale le operazioni di normale manutenzione del contenuto del media.

Il "Sistema Operativo" in dotazione è appena sufficiente e non offre grandi prestazioni. Una novità che merita menzione è la possibilità di proteggere un file da sovrascrittura o cancellazione.

Ogni disco può essere etichettato da un nome (massimo 10 caratteri senza spazi). L'uso della label è addirittura più avanzata di quanto abbiamo visto per un decennio usando il DOS del PC. I comandi di salvataggio possono usarla in alternativa all'indicazione numerica dell'unità come ad esempio:

SAVE DSK.MYDISK.MYPROGRAM

Piuttosto che

SAVE DSK1.MYPROGRAM

Nel primo caso il sistema individua da solo il drive con il nome corretto, nel secondo esempio siamo noi che indichiamo il drive 1 come destinatario del comando.

Altre convenzioni sull'utilizzo della periferica a disco sono abbastanza normali: i nomi dei file possono essere lunghi al massimo 10 caratteri senza spazi e altri segni di interpunzione.

Interessanti le possibilità di gestione di file di record come archivi su disco. Anche se limitati in dimensioni (massimo 128 record per file) sono sicuramente una novità di utilizzo e di istruzione per l'utente che vuole fare un salto di qualità e passare dalle semplici istruzioni BASIC ad una gestione di archivi; un passaggio questo che fa la differenza fra un principiante e un utente evoluto nella programmazione di calcolatori.

La TI ha previsto modalità di trasferimento di dati dai moduli ROM alle cassette/dischi implementando una istruzione (OLD nella fattispecie) che trasferisce il contenuto ndella ROM nell'area RAM interessata alle istruzioni di salvataggio.

Conclusioni

Per le scelte commerciali della texas Instruments, anche il TI 99/4A ha rischiato di essere presto dimenticato. Per fortuna non è stato così e gli appassionati, utilizzando anche l'inventiva dei costruttori esterni, hanno portato avanti per molti anni il sistema arricchendolo come pochi altri.

Anche oggi si trovano gruppi attivi un po' in tutto il mondo e si possono acquistare periferiche e sistemi "clone" con relativa poca spesa.

In conclusione un sistema sicuramente all'avanguardia al momento della sua uscita che purtroppo ha dovuto lottare ad armi impari contro avversari superdotati come il Commodore e lo Spectrum. Ancora una volta il mercato ha scelto le soluzioni meno costose anche se meno totate di potensialità. Così va il mondo...

[Tn]

Come eravamo...

Febbraio 1983

Bit n. 36 febbraio 1983

Siamo già al sesto anno di vita per la rivista Bit della Jackson. Nel numero 36 del febbraio 1983 presenta un calcolatore tutto italiano ma non il risultato del lavoro di una grande ditta in grado di investire capitali in ricerca e sviluppo, frutto invece dell'ingegno e passione di un piccolo gruppo di amici (appena tre) che in quel di Trento "inventano" un sistema Z80 con capacità e soluzioni da far invidia a molti sistemi più blasonati. Alcune pagine della prova hardware nella pagina a fronte.



Anno 6 - n° 36 - Febbraio 1983

In copertina:
Black Star
Fotografia: Photo-Ini

Sommario

Il sommario della rivista presenta è ricco di spunti interessanti. Prima di tutto la presentazione dell'Apple Lisa al Consumer Electronic Show di Las Vegas. Se ne sa ancora poco ma è chiaro per tutti che questa è una macchina "diversa". Interessante anche la panoramica sulle soluzioni "portatili" anche se siamo lontani da una vera fruizione del computer "on the road", l'idea di portarsi in giro il proprio fedele compagno "di giochi", comincia a farsi strada.

Bitest:
Black Star

PENTASYSTEM: ECHI DALLA PROVINCIA INGEGNOSA

Bitest:
Black Star

Bitest:
Black Star

Retro Riviste

La rassegna dell'editoria specializzata dai primi anni '80 ad oggi

Xelectron



Xelectron è l'opuscolo dedicato all'informatica della rivista CQ Elettronica.

Il numero del quale potete ammirare la copertina in questa pagina è del novembre 1983 ma mi mancano dati precisi sulla pubblicazione. Ad esempio quando essa sia iniziata e quanti numeri sono usciti. Immaginando che la sua storia sia più o meno analoga a quella di altre pubblicazioni simili, perciò ipotizzo che sia uscita a cavallo degli anni '83-'84 o poco oltre.

Le dimensioni del fascicolo sono "a mezza pagina" (circa un A5), la stampa in bianco/nero, eccetto la copertina e la consistenza di circa 90 pagine.

CQ Elettronica è una rivista storica di hobbysmo elettronico che ha iniziato molto presto (ben prima dell'80) a pubblicare articoli sui microprocessori, mentre il mondo d'elezione della rivista è quello delle onde radio, come si deduce anche dal nome della testata.

Su CQ si è "fatto le ossa" quel tal ingegner Becattini di Firenze che fonderà la General Processor, una delle prime ditte in Italia a progettare un microcomputer per scopi professionali.

In effetti questo fascicoletto sembra sia la pubblicazione privata del Becattini, visto che sua è la firma in calce a gran parte degli articoli. Non possiamo dire che sia stato sempre così, purtroppo di questa pubblicazione possiedo questo unico numero, recuperato fortunosamente in un mercatino di radioamatori.

Leggendo la cronistoria dell'informatica personale non si ha la vera misura di quel fermento che ha caratterizzato il quinquennio d'oro (1980-85), nel senso che si potrebbe facilmente pensare che l'uscita del PC IBM avvenuta nel 1981 abbia da subito ammazzato il mercato. Naturalmente non è così e si vede anche da queste pubblicazioni come la fase pionieristica

Scheda

Titolo: *Xelectron*

Sottotitolo: Supplemento a CQ elettronica

Editore:

Web:

Lingua: *Italiano*

Prezzo: *Lire 2.000*

Pagine: *90*

caratterizzata dai kit e dall'auto-costruzione, sia durata per anni e abbia convissuto tranquillamente con le macchine di serie professionali (o vendute come tali).

Gli articoli ospitati sono tutti ad un livello più che dignitoso, al punto da risultare godibili anche al giorno d'oggi (per un retro computerista, ci si intende!). Si deduce che sta imperversando un kit chiamato G5, basato su Z80 e spiegato un po' sulla rivista "madre" CQ e un po' approfondito nel fascicoletto Xelectron. Brutto vizio questo di far saltare il lettore da Erode a Pilato obbligandolo a comperare due riviste al posto di una, ma tant'è il marketing è marketing.

In ogni caso mi sembra di poter affermare che per i possessori di questo G5 (la G sta sicuramente per "General"), non potevano rinunciare a questo supplemento, se volevano ricavarne il meglio dalla loro costruzione.

Rendiamo giustizia anche al rimanente contenuto che si occupa di altri personal, come il progetto di una penna ottica per Spectrum, o la discreta presentazione del New-Brian, uno dei micro di derivazione BBC che ha avuto i propri estimatori anche in Italia. Altro micro "serio" è il Micro Professor III della Microtech, che merita una preview sulla rivista.

Perfino gli articoli "terra-terra", come il classico algoritmo in Basic per il calcolo dei bioritmi, sono di piacevoli lettura, forse per una particolare attenzione della rivista ad andare sul concreto ma senza far mancare tutte le necessarie informazioni al lettore.

Conclusioni.

In definitiva una pubblicazione dal costo forse non proprio contenuto (con 2.000 lire nel 1983 si poteva mangiarsi una pizza), ma che vale per ogni pagina di stampa. Peccato non avere oggi a disposizione la collezione completa, ma non dispero che qualche appassionato un giorno o l'altro non possa deliziarci con la scansione di tutti i volumi e la loro disponibilità su Web.

[Sn]


XELECTRON

SUPPLEMENTO CQ ELETTRONICA

novembre 1983

sommario

IL MONITOR

QUALCOSA DI PIÙ SULLA STRUTTURA INTERNA

G5 e il LINGUAGGIO MACCHINA

Una ECONOMICA TASTIERA per il vostro MICROCOMPUTER

INTERFACCIA per STAMPANTE

INTERFACCIA per REGISTRATORE

— XELECTRON 11/83 —

A fronte la copertina Sommario e colophon

TAMC

Teoria e Applicazioni delle Macchine Calcolatrici

Precisione infinita

Basta una infarinatura di informatica per sapere che i calcolatori non fanno i calcoli con precisione infinita. Ad esempio il numero di cifre decimali rappresentabili sono limitate a quanto implementato nel linguaggio di programmazione, cioè a come i progettisti hanno deciso di rappresentare le variabili in memoria.

Normalmente non si va oltre le 12/15 cifre decimali. Gli stessi spreadsheet, per loro natura deputati al calcolo, si fermano normalmente oltre questo limite. Ad esempio Excel (versione 2004 su OS X) fornisce al massimo 16 cifre dopo la virgola, poi aggiunge perissecuamente degli zeri se si ordona un formato numerico con più decimali. La stessa "Calcolatrice", classico accessorio da scrivania, presenta la stessa limitazione e non dubito che anche su altri sistemi (anche se onestamente non ho provato) le limitazioni siano più o meno le stesse.

-"E allora?" Direte voi -"non ti bastano 16 cifre dopo la virgola, cosa diavolo devi calcolare con maggiore precisione?". Beh, ad esempio la traiettoria di un veicolo spaziale, il decadimento di un isotopo radioattivo, un sacco di altre cose per non parlare della ricerca sui numeri e in fisica in generale.

Il punto è: come si fa a calcolare un numero di cifre decimali a piacere? Per i cosidetti numeri razionali, quelli che si ottengono con la divisione fra due numeri interi, esiste un algoritmo molto semplice di pochi statement che si basa sulla tecnica che ora andiamo a descrivere.

Possiamo rappresentare il problema nella maniera seguente:

Siano a0, a1, a2,... ak le cifre che rappresentano il numero risultato dell'operazione di divisione dei due numeri interi m e n:

$$\frac{m}{n} = a_0,a_1a_2a_3...a_k$$

a0 è la prima cifra (unità), seguono le altre cifre decimali fino alla k-esima cifra con k maggiore o uguale a uno.

	A	B
16		
17	1 / 19 =	0,052631578947368400
18		
19		

Excel su MAC si ferma alla 16^ cifra decimale significativa. Gli zeri successivi sono un suo artefatto e non riguardano il risultato dell'operazione di divisione mostrata.

Come ci hanno insegnato nei primi rudimenti dell'algebra: "aggiungendo o sottraendo uno stesso numero ai due membri di una uguaglianza, l'uguaglianza rimane invariata". Possiamo quindi scrivere:

$$\frac{m}{n} - a_0 = a_0{,}a_1a_2a_3...a_k - a_0$$

La cifra a0 è la parte intera della divisione m/n; valore facile da ottenere con qualsiasi linguaggio di programmazione, anche se non prevede esplicitamente l'operatore "parte intera".

Togliendo la parte intera da un numero con decimali se ne ottiene la parte decimale, ovviamente:

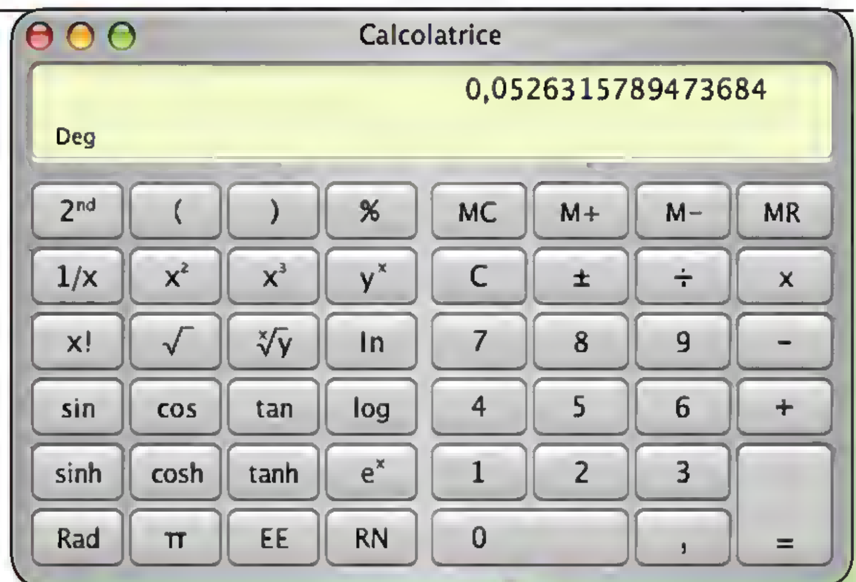
$$\frac{m}{n} - a_0 = 0{,}a_1a_2a_3...a_k$$

Ancora dall'algebra elementare: "Moltiplicando o dividendo i due membri di una uguaglianza per uno stesso numero, l'uguaglianza rimane invariata".

E noi sfruttiamo questo principio per moltiplicare per 10 sia prima che dopo l'uguale:

$$10 \bullet \left(\frac{m}{n} - a_0\right) = a_1{,}a_2a_3...a_k$$

Con semplici passaggi, che confido di non dover spiegare, si ricava che a1 è rappresentabile come:



Anche la calcolatrice da scrivania non va oltre le 16 cifre decimali.

$$a_1 = INT\left(\frac{10(m - a_0 \cdot n)}{n}\right)$$

Si osserva quindi che ricavare la parte intera e la prima cifra decimale si riduce al calcolo di una operazione di parte intera. Un'altra osservazione che facciamo è che questo calcolo ha sempre come denominatore il numero n, mentre cambia il numeratore.

Ovviamente ricavare la seconda cifra decimale è una questione di mera applicazione di questo processo iterativo che si svolge in k passaggi (il numero delle cifre decimali che ci interessano). Situazione ideale per implementare un loop all'interno di un programma con qualsiasi linguaggio di programmazione.

Ad esempio, tanto per rimanere nel tema del retro computing, lo implementiamo in BASIC Applesoft (vedi screen nella pagina successiva), calcolando 1/19 con 60 cifre decimali.

```
]LIST

 10  M = 1
 20  N = 19
 25  K = 60
 30  A =  INT (M / N)
 40   PRINT A",";
 50   FOR I = 1 TO K
 60  M = 10 * (M - A * N)
 80   IF M = 0 THEN  END
 90  A =  INT (M / N)
 100   PRINT A;
 110   NEXT I

]RUN

0,05263157894736842105263157894736842105
2631578947368421052631
]
```

Il listato BASIC Applesoft e il risultato del RUN con precisione a 60 cifre.

Abbiamo quindi dimostrato che i programmi "standard" come l'Excel o la Calcolatrice, forniscono un risultato approssimato. Una osservazione più "sottile" si ricava dalla lista delle cifre decimali.

Notate che la sequenza di cifre è periodica e precisamente di periodo 18. Immagino vi ricordiate tutti della presenza dei numeri "periodici" fin dalle scuole medie. La teoria assicura che i numeri razionali, cioè quelli che si ottengono da operazioni di divisione di due numeri interi, sono periodici e il periodo è sempre inferiore al valore del denominatore cioè a (n - 1).

Per coloro che sono più familiari con una notazione Pascal-like, presentiamo la versione del programma proprio in Pascal, che confesso è un linguaggio che uso spesso per l'implementazione di algoritmi elementari.

Il Borland Pascal, io uso la versione 7.0 in DOS, è ideale: agile nell'uso e veloce nella compilazione e nell'esecuzione è quanto di meglio si possa desiderare per questo tipo di lavori.

Nello screen in fondo a questa pagina la finestra di editing con il programma.

Per rendere più funzionale il programma e per ottenere un certo ordine di stampa che ci permetta di valutare al volo la consistenza del periodo, ho voluto arricchire un po' il sorgente implementando un meccanismo di ricerca automatica del periodo.

Il sorgente di questa versione è riportato nel box della pagina a fronte, dove è presente anche una porzione dell'output ottenuto.

A parte la richiesta di fornire i valori da tastiera e la stampa con impaginazione, il sorgente presenta i suoi punti più interessanti nella ricerca del periodo.

Le cifre calcolate e immagazzinate in un vettore di interi con dimensione 1000, vengono esa-

L'implementazione dell'algoritmo in Pascal (ambiente Borland Pascal 7.0).

```
 File  Edit  Search  Run  Compile  Debug  Tools  Options  Window  Help
=[■]================ \BP\EXAMPLES\CHESS\MULTIP.PAS ================1=[↕]=
Program Multiprecision;

Var
  m, n, k, i, a: integer;

begin
     m := 1; n := 19; k := 60;
     a := m div n;
     write(a, ',');
     for i:=1 to k do begin
         m := 10 * (m - a * n);
         a := m div n;
         write(a);
         end;
     readln;
end._

==== 16:5 ====
F1 Help  F2 Save  F3 Open  Alt+F9 Compile  F9 Make  Alt+F10 Local menu
```

minate a blocchi per cercare il periodo, che si sa essere obbligatoriamente presente.

Attenzione: non è detto che il periodo inizi subito dopo la virgola, possono esserci delle altre cifre decimali non periodiche, ma poi il pezzo periodico parte di sicuro, magari di un solo carattere, come per il numero 1/12 (provare).

Conclusione.

In questo articolo abbiamo usato il calcolatore per rappresentare dei numeri razionali con precisione virtualmente infinita. Inoltre abbiamo verificato l'esistenza della periodicità per le cifre decimali di un numero razionale.

Il computer si può rivelare un utile strumento di indagine delle proprietà dei numeri, come avremo modo di scoprire con gli articoli che seguiranno.

[Sm]

```
Program Multiprecision;

var
   i, m, n, k, a, p, s : integer;
   v : array[1..1000] of integer;

function periodo(r: integer; s: integer): boolean;
begin
     periodo := True;
     for i := s to r do begin
        if v[i] <> v[r+i] then begin
           periodo := False;
           break;
           end;
        end;
end;

begin

   write('numeratore = '); readln(m);
   write('denominatore = '); readln(n);
   write('numero di decimali = '); readln(k);

   (* calcola la parte intera *)
   a := m div n;
   write(a, ',');

   (* calcola le cifre decimali *)
   for i := 1 to k do begin
      m := 10 * (m - a*n);
      a := m div n;
      v[i] := a;
      end;

   (* calcola il periodo *)
   p := n - 1;
   s := 1;

   while not periodo(p, s) do begin
      p := p - 1;
      if p = 0 then begin
         p := n - 1;
         s := s + 1;
         continue;
         end;
   end;

   (* stampa il risultato *)
   for i := 1 to k do begin
      if ((i mod p) = 0) then begin
         write(v[i]);
         writeln;
         write('  ');
         end
      else
         write(v[i]);

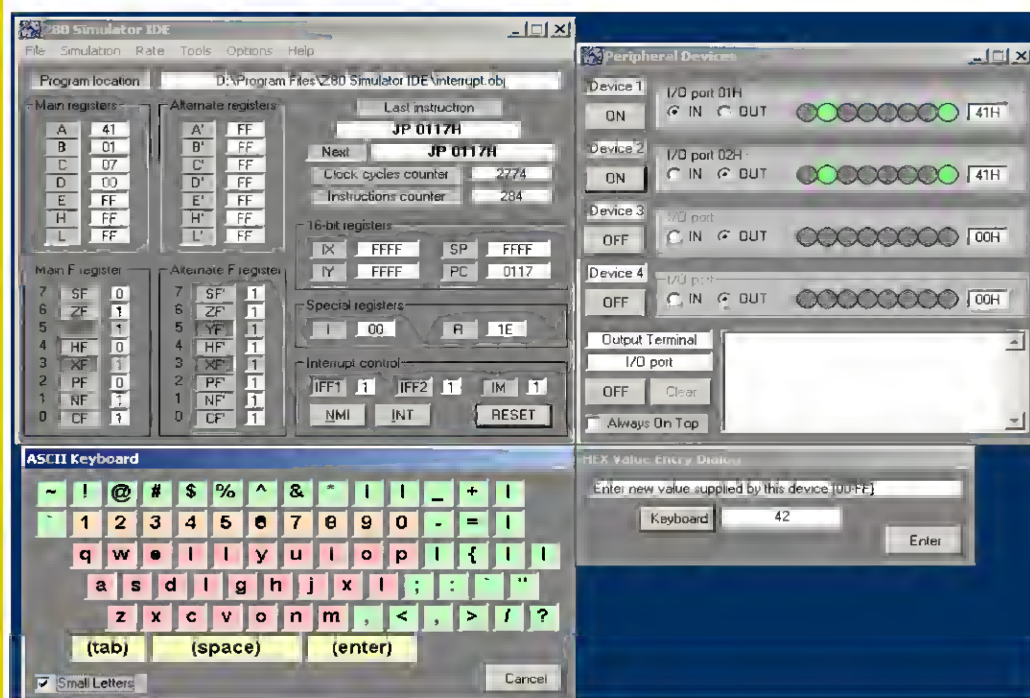
      end;
   readln;
end.
```

```
numeratore = 1
denominatore = 19
numero di decimali = 200
0,052631578947368421
  052631578947368421
  052631578947368421
  052631578947368421
  052631578947368421
  052631578947368421
  052631578947368421
  052631578947368421
  052631578947368421
  052631578947368421
  052631578947368421
  05_
```

Emulazione

Z80 Simulator IDE

I mondi virtuali a volte possono essere molto realistici...



Un ambiente di sviluppo completo per la CPU Z80 con tanto di terminale simulato, controllo dell'I/O e altre comode funzionalità.

Qualsiasi emulatore di computer, console e in generale di qualsivoglia diavoleria moderna che contenga un chip programmabile, ha come base l'emulazione di una CPU. Logico quindi aspettarsi che questo tipo di programmi sia diffusissimo ed infatti è così! Non esiste micro computer o micro controller che non abbia il suo bravo emulatore o simulatore. I due termini simulazione ed emulazione si sovrappongono molto quando l'oggetto è il singolo chip, mentre come sappiamo la differenza è grande nel caso di emulazioni di interi sistemi.

Dunque, dicevamo, reperire un programma di emulazione per una certa CPU non è affatto difficile, anzi ci sono piattaforme per le quali la scelta si amplia notevolmente fino al punto che è addirittura imbarazzante esaminare tutte le possibili soluzioni proposte. Uno di questi casi è sicuramente quello della CPU Zilog Z80, base di tanti micro fortunati negli anni '80 e tutt'ora usatissimo in progetti embedded specifici per i quali conviene affidarsi ad un chip affidabile e molto conosciuto piuttosto che usare altre cose più costose o più ostiche da manipolare.

In questo articolo ho deciso di presentarvi uno dei tanti disponibili: si tratta di un pacchetto di emulazione completo chiamato

Z80 Simulator IDE, scaricabile in versione demo dal sito della ditta che lo commercializza: http://www.oshonsoft.com/. La versione (binari per Windows32) è la 9.95 segno evidente di una continua manutenzione e aggiornamento dell'ambiente che si propone come soluzione completa per l'emulazione della famosa CPU in un ambiente di funzionamento flessibile e completo.

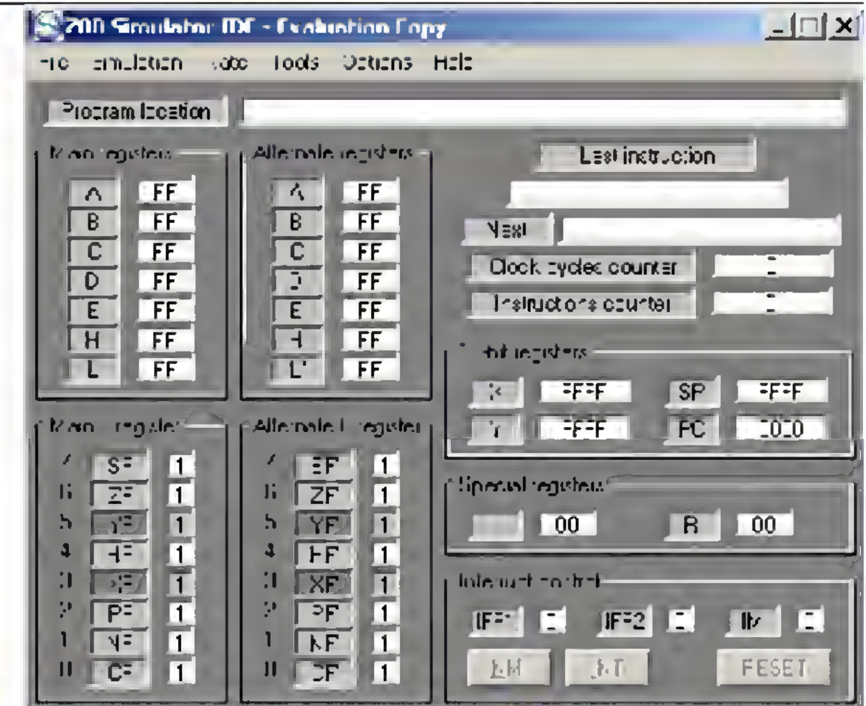


Figura 1 - la main window al lancio del programma.

Installazione

Il programma, per la piattaforma x86 (con Windows XP funziona egregiamente), si installa come qualsiasi classica applicazione Windows e al primo lancio si presenta come in figura 1. In pratica stiamo vedendo la CPU dall'interno con i registri e contatori visualizzati a video, compresa una comoda mnemonica visualizzazione dei registri flag. Come certo sapete la CPU Z80 ha la particolarità che i registri sono doppi all'interno del chip; una particolarità del micro forse più famoso nel settore degli 8 bit, che è stata sempre poco sfruttata. Il set standard e il set di registri alternativi sono entrambi presenti nel programma.

In alto una barra di menù che vedremo è molto ricca di funzioni e che permette di utilizzare alla bisogna l'environment virtuale dove il nostro simulatore è inserito.

La caratterizzazione come IDE è giustificata dalla presenza integrata di un assemblatore/disassemblatore, da un debugger e addirittura da un compilatore BASIC. L'attributo "simulatore" è giustificato dalla precisa temporizzazione con visualizzazione dei cicli di clock durante lo svolgimento del programma. E' possibile ovviamente l'esecuzione step-by-step delle istruzioni, così come l'impostazione di breakpoint per arrestare l'esecuzione ad un certo punto del codice.

Esplorazione

Le prime funzioni (voce File del menù) permettono di pulire per intero i 64Kb di memoria che equipaggiano l'environment, caricare un file e salvare il contenuto su un file a disco.

L'emulazione viene guidata dalle funzionalità raggiungibili attraverso l'apposito menù e che permettono l'esecuzione descritta precedentemente. Una nota particolare merita la possibilità di variare la frequen-



Figura 2 - Lo schema di colori denominato "Blu Berry", decisamente un pugno nell'occhio!

za di esecuzione da un minimo (modalità slow) fino alla modalità "Ultimate" che sfrutta tutto il clock disponibile sul sistema host e spegne il refresh del video durante l'esecuzione.

L'inserimento dei dati, oltre che da un file esterno, può essere fatta caricando direttamente i valori binari nella memoria con un editor appositamente predisposto.

Il tool Disassembler è indipendente dalla memoria del simulatore, dispone cioè di una propria area di memoria e può essere usato senza "sporcare" la RAM del simulatore stesso. E' come disporre di più sistemi paralleli: su uno si realizza il codice, poi lo si trasporta nell'emulatore e lo si prova "al volo".

La gestione dell'I/O è particolarmente indovinata, secondo me. Consiste nella possibilità di visualizzare e modificare all'occorrenza il valore delle singole porte (I/O Port Editor). In aggiunta a questo si possono collegare fino a quattro "terminali virtuali" che presentano il contenuto in ASCII dell'istruzione di OUT sulla porta corrispondente e accettano l'input da tastiera per l'uso dell'istruzione IN della CPU emulata. Durante le nostre prove queste quattro porte virtuali si sono rivelate usabilissime ma purtroppo poche rispetto a quanto avremmo avuto bisogno.

Anche l'Assembler, così come il dissasemblatore, è esterno all'ambiente e permette di editare e assemblare un listato trasferendo poi l'assemblato nella memoria del simulatore. Ovviamente come assemblatore può essere usato un programma esterno (l'autore consiglia il Turbo Assembler di Borland).

Una delle funzionalità che mi è piaciuta più di tutte è la gestione dell'I/O simulato. L'IDE prevede infatti un controllo dei bit in ingresso/uscita dalle porte con la possibilità di fermare l'istruzione IN per ricevere i dati direttamente dalla tastiera e un comodo log di quanto va e viene dalle porte in formato testo registrato a disco.

Un'altra caratteristica particolare permette l'abilitazione delle istruzioni "non ufficiali", cioè quei codici che danno accesso a comportamenti prevedibili da parte della CPU ma che per qualche ragione la Zilog ha deciso che non servivano. Sono comunque istruzioni a due byte il cui uso non è frequente e che comunque possono essere sostituite da una sequenza di istru-

zioni standard. Questo avviene normalmente nei prodotti industriali, dove l'utilizzo di funzioni non standard potrebbe comportare qualche difficoltà di manutenzione e supporto dei progetti. Senza contare che non è affatto certo che produttori diversi dello stesso chip implementino per intero il disegno originale della CPU.

L'autore (certo Vladimir Soso) si è pure levato lo sfizio di dotare il suo prodotto di una certa, se pur abbastanza rudimentale, possibilità di skinning. Gli schemi di colore che si possono scegliere da una lista sono decisamente sufficienti a soddisfare l'esteta più esigente, con qualche combinazione decisamente "esoterica", come la Blue Berry mostrata nella figura 2.

Per questo articolo ci è venuta comoda la versione "Brick Set" , una sorta di "legolandia", per via dei vivaci colori che dovrebbero aiutare nell'individuazione dei campi e dei valori che andremo ad utilizzare.

Non avrebbe molto senso in questa sede descrivere tutte le funzionalità, la cui spiegazione si trova, anche se in forma forse troppo spartana, nell'help allegato al programma. Vediamo piuttosto il comportamento dell'IDE nella pratica.

Uso

Abbiamo scelto di provare la simulazione del codice monitor del micro NEZ80, già noto ai nostri lettori grazie agli articoli ad esso dedicati sulla rivista.

Prima di tutto ci siamo procurati il binario (1 Kb) dal sito www.neZ80.org e l'abbiamo salvato con nome Mon_382.BIN nella directory dell'emulatore.

Proviamo a caricarlo nella finestra del disassemblatore e lanciare il tool (figura 3)

Quello che ottiemo è il codice disassemblato in formato listing che può anche essere salvato su file. Ogni locazione oggetto di salto viene etichettata con numero progressivo e al termine del listato una comoda tabella di riassunto ce ne fornisce la mappa.

E' necessario sapere che nel

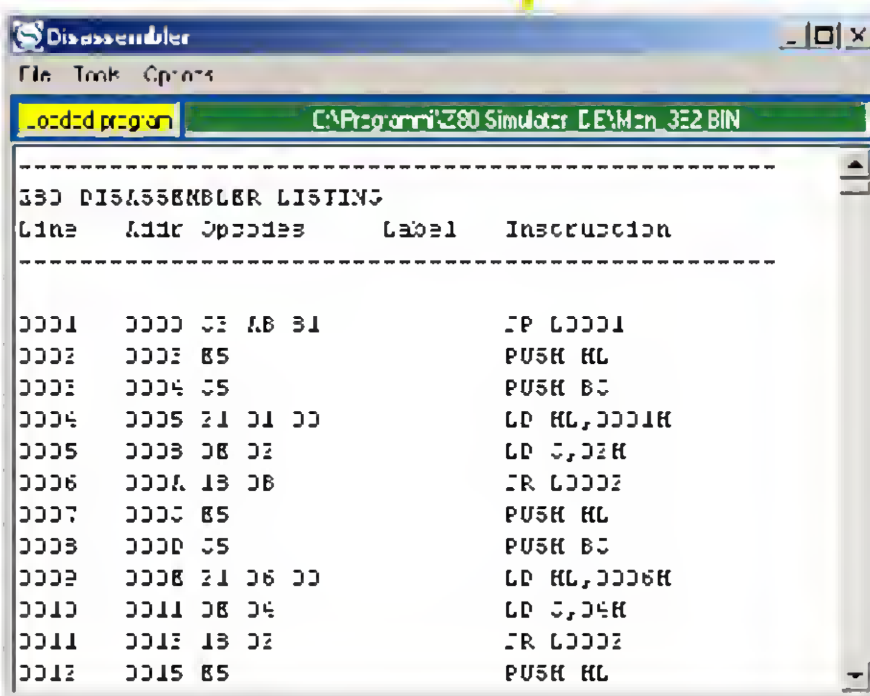


Figura 3 - Il disassemblato del monitor 382 del micro NEZ80.

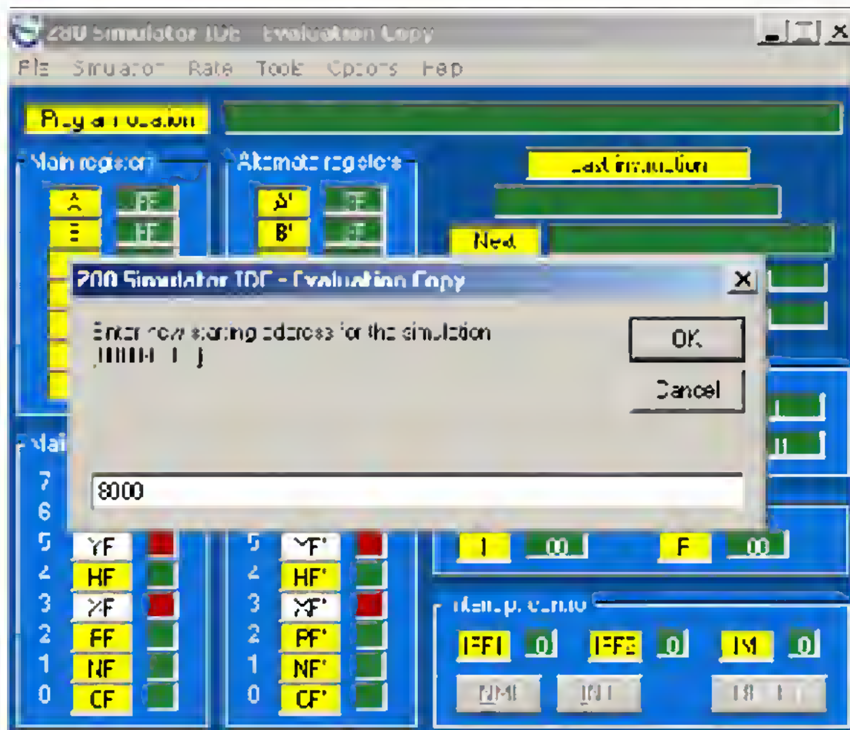


Figura 4 - Caricamento dell'indirizzo di partenza nel registro Program Counter.

progetto di Nuova Elettronica questo k di codice sta nelle locazioni dalla 8000H alla 83FFH (valori esadecimali).

Qui abbiamo trovato una carenza del programma Z80 Simulator IDE perchè ci è stato impossibile caricare il codice dalla locazione di memoria desiderata, anche settando la locazione di partenza, che evidentemente ha effetto solo sul valore del program counter. Alla fine abbiamo aggiunto degli zeri binari in testa alle istruzioni del monitor in modo da riportare le vere istruzioni dello stesso alla locazione corretta. Così abbiamo scoperto che anche il salvataggio della memoria agisce come un dump stupido degli interi 64Kb. Una scelta accettabile e tutto sommato che semplifica le operazioni di load/save, basta saperlo!

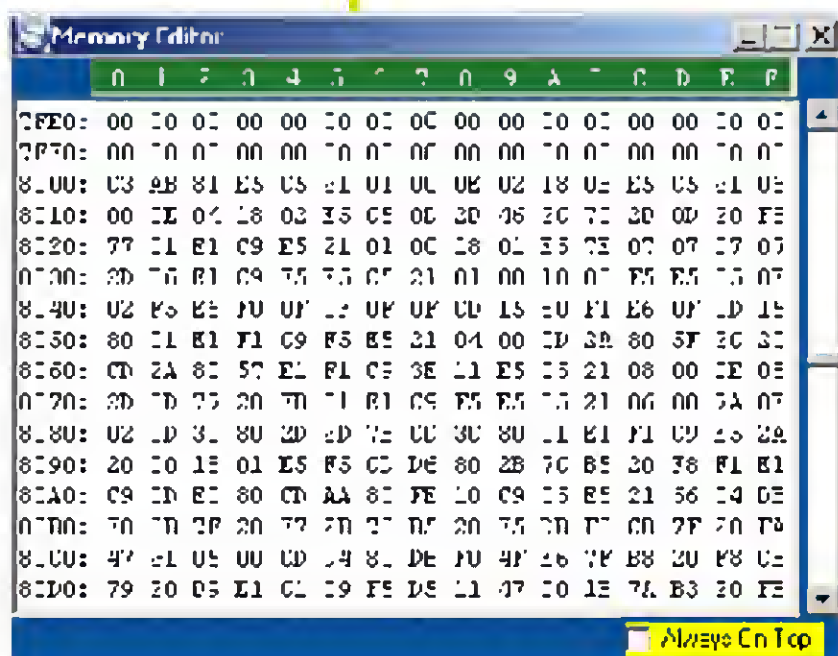
La scelta della locazione dalla quale far partire il programma viene fatta con la funzione "Change starting address" disponibile dal menù Options.

Come si vede il file risulta caricato e il PC (Program Counter) è settato all'indirizzo corretto, come una rapida visualizzazione con il memory editor ci conferma (figure 4 e 5)

Per quanto riguarda le porte di output, esse sono costituite da otto display a sette segmenti corrispndenti al range di indirizzi da F0h a F7h. Alla porta F0h è associato anche l'input della tastiera esadecimale. Il micro Z80 ha uno spazio di indirizzamento di 64 byte per l'I/O ed ognuno degli indirizzi può essere associato contemporaneamente ad un dispositivo in Output e di Input.

Volendo controllare le porte nel simulatore andremo ad agire sull'opzione "Peripheral Device" inserendo gli indirizzi da controllare e la direzione del flusso (IN o OUT).

Figura 5 - Il codice binario del monitor 382 caricato a partire dalla locazione 8000h, mostrato nella finestra del Memory Editor.



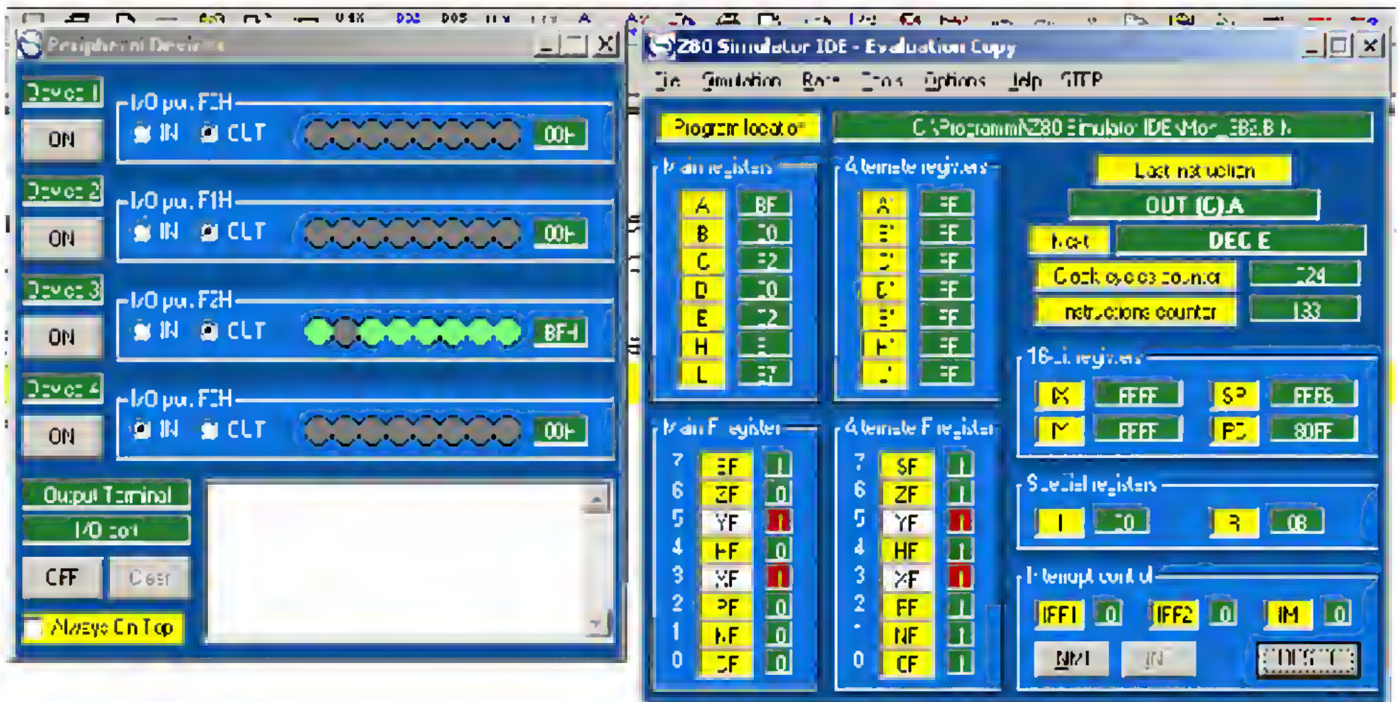


Figura 6 - Il monitor delle porte di I/O. Si noti la porta F2h interessata a bit di uscita settati al valore 1 (led verde acceso).

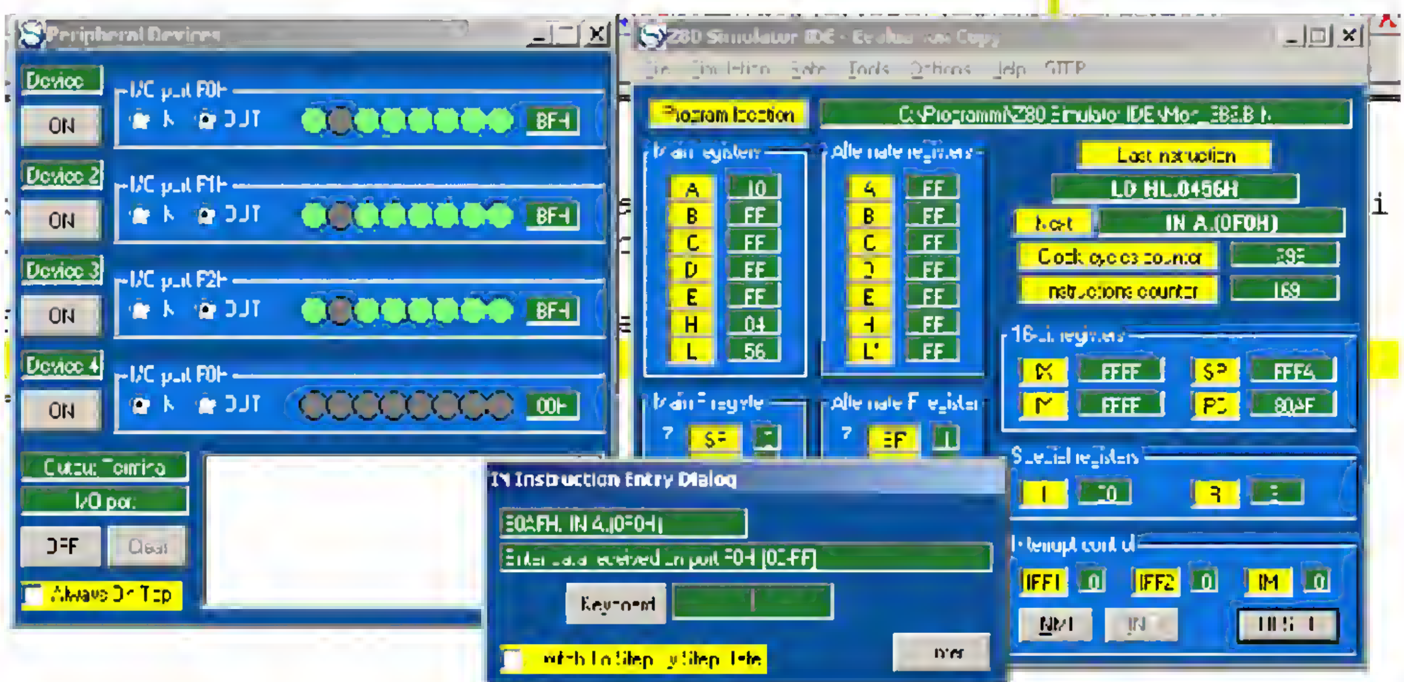
Purtroppo il numero di device gestibili è limitato a quattro, ma rimane comunque l'opzione "I/O Editor" per visualizzare l'intero range disponibile per le funzioni di I/O del processore Z80 (da 00h a FFh).

L'esecuzione a singolo step è quella che aiuta meglio a capire il funzionamento di un nuovo sistema. Si esegue impostando il "Rate" a "Step by Step" e facendo partire l'emulazione (tasto F1). Volendo si può attivare il log, che registra tutte le operazioni fatte, istruzione per istruzione e il log dell'I/O che registra i byte che transitano per le porte dei device.

Con F2 si effettua un singolo step di esecuzione.

In figura 6 siamo arrivati alla prima istruzione di output che deposita sulla porta F2 il valore binario BF. Come si vede dalla figura il monitor dei device ci rende il risultato dell'operazione.

Figura 7 - Il programma in esecuzione sta attendendo un input da associare alla porta F0h. di ingresso.



Nella figura 7 invece siamo all'istruzione di input, inseriamo il valore 0A e vediamo cosa succede...

La visualizzazione del log di esecuzione è veramente molto dettagliata (figura 8) e presenta il valore di tutti i registri, compresi quelli alternativi e le due parole di stato a livello di bit.

Il compilatore BASIC

Come abbiamo accennato all'inizio dell'articolo, il programma include un compilatore Basic che permette di creare un file binario usando una sintassi decisamente più comoda rispetto all'assembler. Ovviamente il Basic non può sostituire in tutto e per tutto un buon assembler, ma per qualche routine di test potrebbe rivelarsi un aiuto prezioso.

Abbiamo compilato un programmino di test che è il seguente:

```
Dim i As Integer
Dim a As Integer
For i = 1 To 10
a = a + i
Next i
```

ottenendo dalla compilazione un file .asm che è il seguente:

```
; Begin
        LD IX,0FF00H
        LD SP,0FEFCH
; 1: Dim i As Integer
;          The address of 'i' is
IX-2 (FEFEH)
; 2: Dim a As Integer
;          The address of 'a' is
IX-4 (FEFCH)
; 3: For i = 1 To 10
        LD HL,0001H
        LD (IX-02H),L
        LD (IX-01H),H
L0001:          NOP
        LD L,(IX-02H)
        LD H,(IX-01H)
        LD DE,000AH
        CALL C001
        JR Z,L0003
        JP NC,L0002
L0003:          NOP
; 4: a = a + i
        LD L,(IX-04H)
        LD H,(IX-03H)
        LD E,(IX-02H)
        LD D,(IX-01H)
        ADD HL,DE
        LD (IX-04H),L
        LD (IX-03H),H
; 5: Next i
        LD L,(IX-02H)
        LD H,(IX-01H)
        INC HL
        LD (IX-02H),L
        LD (IX-01H),H
        JP L0001
L0002:          NOP
; 6:
; End of program
        HALT
; Integer Comparison Routine
C001:   LD A,H
        XOR D
        JP P,C002
        XOR H
        AND 80H
        CP 7FH
        RET
C002:   SBC HL,DE
        RET
; End of listing

         .END
```

log.txt - Blocco note

File Modifica Formato Visualizza ?

Simulation started at 2[illegible]/07/2007 [illegible].49.

No.	PC	Instruction	A	SZ-H-PNC	B	C	D	E	H	L	IX	IY	SP	A'	SZ-H-PNC'
1	8000	JP [illegible]	FF	[illegible]	FF	FF	FF	FF	FF	FF	FFFF	FFFF	FFFF	FF	[illegible]
2	80AB	LD HL,0000H	FF	[illegible]	FF	FF	FF	FF	00	00	FFFF	FFFF	FFFF	FF	[illegible]
3	80AE	LD (0022H),HL	FF	[illegible]	FF	FF	FF	FF	00	00	FFFF	FFFF	FFFF	FF	[illegible]
4	[illegible]	DEC L	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	00	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]
5	[illegible]	DEC H	FF	[illegible]	FF	FF	FF	FF	FF	FF	FFFF	FFFF	FFFF	FF	[illegible]
6	80B3	LD A,(HL)	00	[illegible]	FF	FF	FF	FF	FF	FF	FFFF	FFFF	FFFF	FF	[illegible]
7	80B4	CPL	FF	[illegible]	FF	FF	FF	FF	FF	FF	FFFF	FFFF	FFFF	FF	[illegible]
8	80B5	LD (HL),A	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]
9	[illegible]	CP (HL)	FF	[illegible]	FF	FF	FF	FF	FF	FF	FFFF	FFFF	FFFF	FF	[illegible]
10	80B7	JR NZ,$-05H	FF	[illegible]	FF	FF	FF	FF	FF	FF	FFFF	FFFF	FFFF	FF	[illegible]
11	80B9	CPL	00	[illegible]	FF	FF	FF	FF	FF	FF	FFFF	FFFF	FFFF	FF	[illegible]
12	80BA	LD (HL),A	00	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]
13	[illegible]	INC HL	00	[illegible]	FF	FF	FF	FF	00	00	FFFF	FFFF	FFFF	FF	[illegible]
14	80BC	LD SP,HL	00	[illegible]	FF	FF	FF	FF	00	00	FFFF	FFFF	0000	FF	[illegible]
15	[illegible]	LD HL,0066H	00	[illegible]	FF	FF	FF	FF	00	66	FFFF	FFFF	0000	FF	[illegible]

Figura 8 - Il comodo LOG che registra tutto quello che succede.

Ora possiamo passare per l'assemblatore per ottenere il file .obj contente il codice oggetto (figura 9) e quindi caricarlo in memoria e farlo partire.

Il basic in dotazione ha delle limitazioni ma appare comunque all'altezza del compito assegnatogli, che è quello di assemblare semplici routine avvalendosi di pattern predisposti da esperti e che utilizzano, come abbiamo potuto constatare di persona, in maniera intensiva i registri di indirizzamento IX e IY.

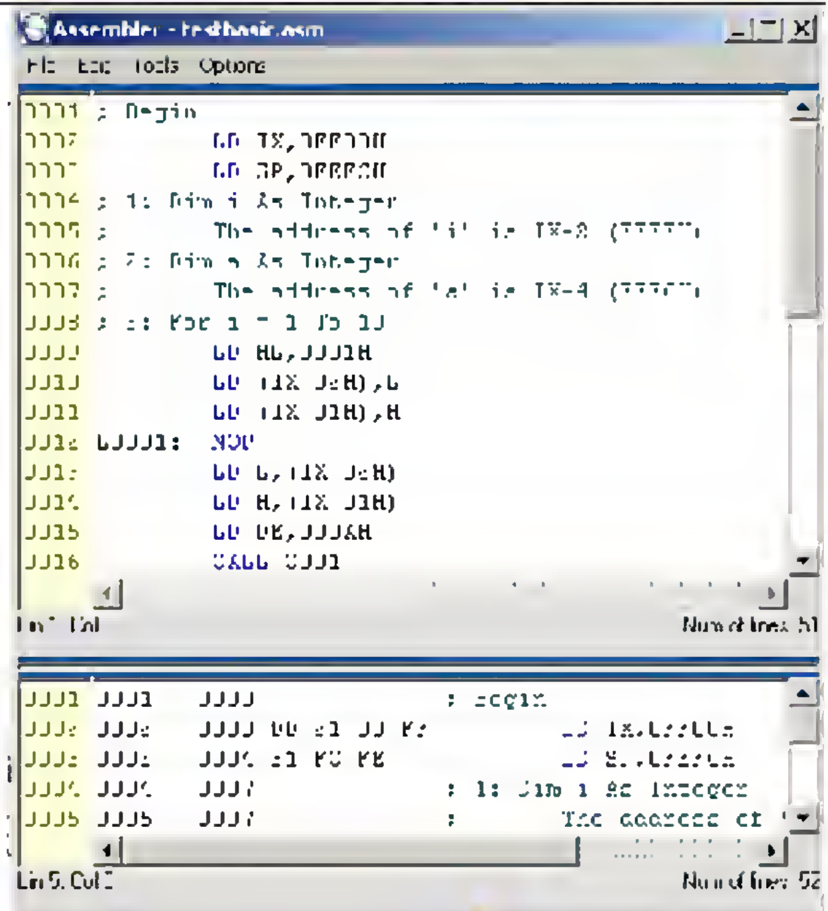


Figura 9 - Stiamo assemblando il codice oggetto ottenuto dal compilatore BASIC.

Conclusioni

Il programma Z80 Simulator IDE della Oshon Software, a dispetto di qualche piccolo difetto, è un buon esempio di come si dovrebbero realizzare simili tool: semplici nell'utilizzo e completi nelle funzionalità.

La valenza didattica è sicuramente il lato dove meglio si esprime il progetto, mentre per un uso professionale forse manca di qualche agevolazione per aiutare lo sviluppatore a tenere traccia dei vari pezzi di codice che sta via via assemblando. C'è comuinque da riconoscere che un hardwarista di professione non si affiderebbe certo ad un tool così "leggero" e sceglierebbe un vero emulatore "in hardware" della CPU che stà utilizzando.

[L2]

Una visita a...

Obsolete Computer Museum

Visite virtuali o reali nei luoghi di maggiore interesse per l'hobby del retrocomputing

XML

Obsolete Computer Museum

Welcome to the Obsolete Computer Museum!

Some Decisions

I've come to some decisions about the future of the Museum. First off, I wasn't clear where the monetary cost was occurring. The site itself doesn't cost much, in terms of hosting fees. The cost is in physically housing the computers. I rent a climate-controlled storage unit for the majority of the collection, and that's what costs the most. So, the many offers of site hosting are welcome, but not where the financial burdens lie.

It was clear from the messages I've received that most folks don't really care whether I physically have these computers. It's the site that's important. So, donating the physical computers isn't really a problem. I can do that and still keep the Museum without horribly upsetting anyone. And I do have some leads. My preference is to give them to other museums that would physically show them to the public, rather than to private collectors. So I'm going to start down that route, slowly. It will also give me a chance to take new photos, with a good digicam.

Message boards! Yep, I've set up message boards. They're in eternal BETA test. **Try 'em out!**

Both the Helpline and the Comments system are now also accessible as **RSS 2.0** news feeds. Find yourself a **newsreader program** and point it at:

- http://www.obsoletecomputermuseum.org/helpline.rss
- http://www.obsoletecomputermuseum.org/comments.rss

3D models:

- **TRS-80 Model 100**
- **IMSAI 8080**
- **MITS Altair 8800**

(They require a Java-enabled browser.) Other suggestions? Let me know at **computer.museum@gmail.com**.

Here's my personal site, **tom.carlson.org**, with (awful) music, (self-congratulatory) health information, (goofy) pictures of me, and (pinko left-wing) political rantings.

All'indirizzo http://www.obsoletecomputermuseum.org/oldindex.shtml si trova la vetrina di una delle tante iniziative museali rivolte alla conservazione delle macchine di calcolo dall'origine dell'informatica ai nostri giorni. Penso che tutti i retro computeristi hanno la volontà di mostrare la loro collezione e quale migliore vetrina del Web per raggiungere il mondo intero?

Come quasi tutte le altre iniziative del genere, anche Obsolete Computer Museum si basa sul lavoro e sulla collezione privata di un appassionato, certo Tom Carlson, che dispone di un numero di pezzi davvero notevole (più di 100 modelli diversi).

Ho deciso di presentare questa iniziativa non perché non esistano anche in Italia analoghi siti e iniziative, che abbiamo in programma di recensire comunque pian piano, ma perché visitandone il sito non tanto tempo fa, ho raccolto una certa difficoltà dell'autore che prelude ad un ridimensionamento dell'iniziativa.

Cosa è successo? In pratica quello che dobbiamo aspettarci per tutte quelle iniziative che, cresciute notevolmente, vorrebbero presentarsi al mondo, non fosse altro che per dare uno scopo a tanta fatica. Il problema è di scala: un conto è raccogliere materiale qua e là stipandolo nel garage o in cantina e un'altro è operare per una vera fruizione della collezione reale. Servono spazi, accessi pub-

XML

Altair 8800

Donor: Don Miller
Location: Williamsburg, VA

Due to the saint-level kindness of Don Miller, the Museum now physically contains an Altair 8800. As near as I can tell, it seems to work. The A3 LED isn't working, but everything else seems to flash appropriately.

- **Altair Front Panel** - Just the front panel, in dim light so you can see the lights. Notice that the A3 LED isn't working.
- **Altair Front** - View of front and top in dim light.
- **Altair Front** - View of front and top in strong light.
- **MITS Logo** - The legendary MITS Logo in dim light. The colors are wrong, but you can see the actual logo better.
- **MITS Logo** - The same logo, under strong lighting. The colors are correct, but it's hard to make out the MITS logo.
- **Backside of the Front Panel** - This is the back side of the board holding all those toggle switches and LEDs. A ribbon cable connects this panel to the processor card.
- **Back of Case** - Jacks for a tape drive and a TTY terminal are included. There is also a video jack, but it's not actually connected to anything.
- **Power Supply** - Now that's a power supply! Watch your fingers, there's nothing protecting the ignorant here.
- **S-100 Bus** - Every card, from the CPU, to memory, to the I/O plugs in here. Notice the nice plastic stabilizing bars on either side. I've seen other Altairs that lack these, so I don't know if they are official MITS stuff or not.
- **S-100 Bus Cards.** These cards comprise the actual computer. The rest of the case justs hooks them together and supplies the power.
  - **CPU Card** - The white chip is the 8080. These S-100 bus computers did not have a motherboard like current PCs. The

La home page del sito e a fianco una delle schede tecniche disponibili.

blici, servizi, tempo e risorse. Tom si lamenta infatti di non riuscire a coprire le spese di stoccaggio del materiale e men che meno i costi di una eventuale esposizione. Il risultato è un appello alla donazione lanciato sul Web con la "minaccia" di una chiusura dell'iniziativa. Per questo motivo ho deciso di fissare il tempo e fotografare lo stato di fatto di quello che è uno dei più godibili siti di collezione.

Obsolete Computer Museum (OCM) si presenta abbastanza spartano nella grafica e sobrio nell'uso dei colori. La home page è semplicemente un testo strutturato dove trovano posto le comunicazioni, le news, l'elenco delle macchine e i link alle iniziative di nuovo varo. Ovviamente l'attenzione viene catturata dalla lista dei sistemi posseduti, davvero lunga e significativa. Troviamo delle macchine decisamente rare come il MITS Altair 8800 e l'IMSAI 8080 (quello di War Game), accanto a modelli che tutti possediamo, basti citare il classico Commodore 64.

La collezione copre praticamente tutta la produzione di home computer anni 70-80 prodotti negli States, mentre manca di quei pezzi particolari che hanno avuto diffusione locale, come kit e macchine prodotte in Italia ad esempio.

L'autore si dichiara pronto ad accettare donazioni di macchine nuove (uno degli scopi della vetrina Web è proprio lo stimolare eventuali donatori), ma anche solo foto e documentazione di sistemi non ancora presenti ma che possono essere significativi e comunque utili in futuro.

Una sezione interessante è quella della scansione 3D delle macchine (per ora solo tre). Si tratta in pratica di un applet Java che consente di ruotare con il mouse la rappresentazione tridimensionale dei sistemi. In questo modo si possono osservare da tutti gli angoli, perfino da sotto!

Curatissime le schede tecniche dei singoli pezzi, davvero una pietra di paragone in termini di completezza ed accuratezza (vedere quella dell'Altair 8800 nella pagina a fronte).

Dal punto di vista interattivo i visitatori possono postare dei commenti relativi ai singoli oggetti, alla stregua di un forum di discussione (poco frequentato per la verità).

In conclusione un punto di riferiemento per tutti i retro computeristi che incessantemente spazzolano il Web alla ricerca di siti e iniziative di scambio di esperienze. Speriamo solo che l'autore riesca a tenere duro rispetto ai suoi propositi di rinuncia e che il sito e le macchine stesse rimangano disponibili per sempre. ***[Tn]***





Due proiezioni dell'Altair 8800 ottenute tramite l'applet Java descritto nell'articolo.

Sul sito anche una aggiornata raccolta di link dei siti (oltre 1600) che linkano le pagine del museo.

Come eravamo...

La storia dei sistemi e degli uomini che hanno creato un mondo nuovo.

I primi calcolatori italiani

Come paese industrializzato l'Italia dal dopoguerra vide la nascita di iniziative commerciali tese a fornire un prodotto italiano di elaborazione dati. I tentativi nascono da lontano e già attorno alla metà degli anni '60 la Ing. Carlo Olivetti era in grado di presentare a livello internazionale il suo "Programma 101", un progetto tutto interno nato dalla passione di un pioniere quale l'ing. Giorgio Perotto deve definirsi.

La sua "Perottina" era una macchina enorme per le dimensioni cui siamo abituati oggi, ma all'epoca concentrava in sé il meglio della tecnologia in uno spazio operativo contenuto. La vendita di 40.000 esemplari (stimata), non convinse comunque l'azienda a proseguire con la decisione necessaria nel business dell'informatica che riprese solo sul finire degli anni '70 quanto le sue macchine calcolatrici cominciarono a ibridarsi con soluzioni elettroniche.

La Olivetti poteva avvalersi di collaborazioni internazionali assolutamente di primissimo ordine nel campo della ricerca e dell'ingegneria informatica. Collaborazioni che le permisero di entrare alla grande nel mondo dei mini computer scegliendo un settore, quello delle aziende di piccola dimensione, con buone capacità di spesa e con la voglia di fidarsi di soluzioni italiane piuttosto che confrontarsi con sistemi anglosassoni che forse prevedevano una qualche conoscenza tecnica superiore e certamente l'uso della lingua inglese, scoglio prevedibilmente ostico per il settore amministrativo aziendale.

Nascono così soluzioni "midrange" come l'S6000, affiancato a soluzioni elaborative di classe personal come il P6060 seguito poi dall'M20 e successivamente dal fenomeno M24.

Sulla nascita dell'S6000 esistono pagine di assoluta poesia scritte dagli stessi autori dell'impresa.



La "Perottina" Olivetti, poi chiamata "Programma 101". Un esemplare è esposto al Museo of Modern Art di New York come esempio di design e di innovazione.

Nomi quali Marisa Bellisario, Elzerino Piol e molti altri si possono riconoscere come i preculsori di una "via italiana" all'informatica, successivamente svenduta quando in azienda entrarono i finanzieri relegando l'ingegneria a ruolo di comprimaria. E invece erano proprio le "trovate" elettro-meccaniche a fare dei prodotti italiani una identità riconoscibile. Certe soluzioni di jubox con dischi rigidi, evidentemente derivate dalla meccanica di precisione cui era capace l'azienda, offrivano soluzioni ingegnose a problemi concreti, come appunto quello legato alle scarse capacità dei supporti magnetici, problemi pratici che si cominciavano a sentire. L'S6000 ebbe un degno seguito nella così chiamata "Linea 1" cui facevano parte i vari M30-40-60 fino all'M20, un sistema che possiamo definire senza tema di essere smentiti "troppo avanzato per i suoi tempi".

Indubbiamente la scelta del processore Z8000 fu un azzardo, con il senno di poi, ma chi poteva immaginare che la vincesse Intel la lotta per la supremazia di mercato nei microprocessori, con quel suo 8086 così astruso nella gestione segmentata della memoria?

Lo Z8000 sembrava essere il successore ideale di uno Z80 che la Zilog stava vendendo "come il pane" e per il quale erano disponibili fior di applicativi e un sistema operativo (il CP/M) che meglio non si sarebbe potuto desiderare. Invece la Intel schiacciò letteralmente la Zilog con il suo peso commerciale ma anche con una visione strategica del mercato che si è rivelata vincente su tutti i fronti. Così il "povero" M20 dovette cedere la mano e con esso anche le innovazioni architetturali della linea 1 con macchine multiprocessori e l'innovativo sistema operativo MOS scritto in Pascal+, idioma che non sopravviverà oltre queste poche implementazioni.

Recentemente ho incontrato un anziano professore universitario (anziano ma di spirito giovane) che mi ha parlato della propria esperienza in Olivetti, condendo il racconto con succosi aneddoti legati alle prime esperienze nei sistemi informatici. Parlare con lui di tante figure lette solo sui libri o nelle cronache più o meno complete dei vari protagonisti, è stata una gioia impossibile da descrivere a parole. Mi sono sentito come un antropologo culturale che batte i paesini sperduti nelle valli dimenticate per raccogliere quelle testimonianze orali. Tesori che sarebbero irrimediabilmente perduti qualora i pochi sopravvissuti, testimoni diretti di un mondo scomparso, non avessero l'opportunità di tramandare.

Il nostro incontro è stato del tut-



Il microprocessore Zilog Z8000, possibile erede della fortunata serie Z80 non ha avuto purtroppo lo stesso successo commerciale.

to casuale, in una città lontanissima dalla residenza di entrambi (la sorprendente, Reggio Calabria, che visitavo per la prima volta). Quando mi parlò del progetto del compilatore COBOL per il sistema operativo COSMOS, ho capito che la casualità di certe situazioni può essere guidata solo da una mano superiore. Subito la mia mente è corsa alle interminabili sessioni di compilazione a lottare (ma con gioia) con quel bug che rendeva del tutto inutili i nostri tentativi di consegnare in tempo quel famigerato programma chiesto dal cliente.

Con l'apice della parabola rappresentata dall'M24, la Olivetti prese velocità lungo la china, purtroppo in discesa, che la portò nel breve volgere di poche stagioni nell'oblio del dimenticatoio.

Mi rendo conto che è facile giudicare gli errori degli altri, soprattutto con il senno di poi. Forse era destino o forse le scelte dei manager sono state comunque le migliori possibili e perlomeno hanno frenato, per quanto era nelle loro umane possibilità, la corsa allo sfacelo. Difficile dirlo ora, nei protagononisti rimane una profonda amarezza e il paradigma della "bella azienda finita miseramente". Mi chiedo come sarebbe stato possibile salvare la società con una impostazione organizzativa ormai del tutto avulsa dalla frenetica corsa al ribasso dei prezzi dell'elettronica di consumo. Certo che altre aziende hanno saputo re-inventarsi e non ci siamo riusciti invece noi italiani, ad onta della nostra sbandierata fantasia e della nostra predisposizione a cavarcela in qualunque situazione.

La Olivetti è certo stato il caso più rilevante nella storia dell'informatica italiana, ma non la sola. Decine di piccole aziende sono nate e cresciute sull'onda dell'entusiasmo dei loro creatori.

Gli anni dal 1978 al 1983 circoscrivono un quinquennio di grande fermento nella ricerca del personal del futuro. Prima della grande stagione degli home ('82-'90), sembrava fosse possibile per ciascuno il potersi ricavare una piccola fetta del mercato dell'elaborazione dati. Soprattutto la piccola e media impresa chiedevano sistemi dal prezzo accessibile ed applicazioni ad-hoc.

Si vede molto chiaramente l'influenza dei primi corsi di elettronica digitale applicata ai processori che consentirono agli ingegneri

L'Olivetti BCS, un'altro progetto di macchina da ufficio che non è considerato un vero calcolatore, anche se era programmabile e presentava gli stessi elementi di un PC: floppy, memoria, stampante, video, tastiera,...



laureati dalle università la possibilità di realizzare in concreto un sogno: quello di costruire la "propria" macchina. Sistemi di sviluppo più o meno sofisticati che andavano raffinandosi di mese in mese, consentivano a questi pionieri l'acquisizione di una padronanza tecnica all'avanguardia.

Nascono così le "micro aziende" di informatica come la Generale Processor a Firenze e la Penta System a Trento. La prima con il suo "Model T" e la seconda con il suo "Black Star" a presidiare un mercato inizialmente poco più grande della regione d'origine, ma entrambe innovative, agili, pronte al cambiamento e sensibili alle esigenze dell'utenza.

Ne ho citato due ma ne avrei potuto citare altre, tutte degne di essere menzionate e recuperate come esperienze di innovazione; le ho scelte perché della loro esperienza rimane più di una flebile traccia, ad esempio sono citate in prove hardware sulle riviste del settore e sulla stampa nazionale hanno comprato spazi pubblicitari, evidente segno di una prospettiva di crescita.

La contingente situazione che si determinò sulla soglia degli anni '80 è derivata da numerosi fattori, tutti probabilmente importanti se non indispensabili per il fiorire di simili iniziative. Va ricordato ad esempio la pubblicazione dei kit sulle riviste di elettronica, veri motivatori dell'ingegno dei primi esploratori del digitale, la disponibilità di documentazione tecnica e degli stessi chip a prezzo abbordabile e discretamente reperibili. Non vanno dimenticate le innovazioni che impegnarono i laboratori universitari e le stesse aziende, in quegli anni disposte ad investire in ricerca.



Il Child Z, primo prodotto industriale della General Processor.

Il caso della General Processor di Firenze è un esempio concreto e anche il più riuscito delle capacità italiane di creare un mercato nazionale.

I tre fondatori: Gianni Becattini, Stefano Giusti e Franco Pirri avevano un background tecnico derivato dai loro studi ingegneristici all'università. L'ingegner Becattini in particolare si era creato una notevole popolarità scrivendo articoli sulla rivista CQ Elettronica con progetti a microprocessore.

Il primo prodotto di successo della GP fu il Child Z (1977), seguito dal Modello T (1979). Il nome voleva richiamare la prima auto prodotta in grande serie da Ford, ma era ripreso anche dalla T della televisione, dalla quale copiava alcune tecniche modulari della produzione in serie. Franco Pirri completò il disegno delle schede elettroniche del Modello T nel Marzo del 1979 e la commercializzazione iniziò subito dopo. Nel 1980 veniva prodotto



Il Model T della GP di Firenze (prima serie), anno di commercializzazione 1979.
Sotto una pagina pubblicitaria per la compagna commerciale.

PERCHÈ ANCHE IL PIÙ ESIGENTE NON HA DUBBI PER SCEGLIERE GENERAL PROCESSOR?

Perché la GP ha più esperienza. La GP è la **prima** azienda italiana ad aver prodotto microcomputers e personal computers; la prima in ordine cronologico e la prima per produttività. È anche la prima per la sua rapida espansione.

Perché la diffusione dei prodotti GP è conferma di qualità. I sistemi GP entrano anche "negli ambienti che contano". Esperti tecnici, istituti universitari, industrie, enti di ricerca (come il Consiglio Nazionale delle Ricerche) si affidano ogni giorno al nome GP.

Perché i prodotti GP sono i più prestigiosi. Ogni progetto è fatto con in mente l'utente finale, i suoi problemi, le sue esigenze. Nessun dettaglio è trascurato e la scelta dei componenti è fatta in base a criteri estremamente rigorosi.

Perché la gamma dei prodotti GP è estremamente vasta: il nuovo Modello T è completamente espandibile in senso verticale:

★ Il T/05 con registratore audio, per l'hobby o per il calcolo scientifico

★ Il T/08 dotato di minifloppy disk per la più vasta gamma di problemi applicativi

★ Il T/10 destinato alla gestione di aziende di medie dimensioni con una estesa memoria a dischetti IBM compatibili

★ Il T/20 con un grande disco da 14 a 24 Mega bytes che vi aspettereste di trovare solo su un sistema di costo molto maggiore

Perché i «personal» della GP dispongono di una delle più vaste biblioteche software del mondo: Il nuovo Modello T è compatibile col famosissimo CP/M (*), il più diffuso sistema operativo a dischi oggi esistente. Sotto il CP/M (*) sono disponibili tutti i più conosciuti linguaggi di programmazione; quindi non più soltanto il BASIC, ma anche FORTRAN, COBOL, APL, PASCAL, BASEX, ASSEMBLER ecc. ecc. Il servizio software della GP è poi a vostra disposizione per personalizzare secondo le vostre necessità i numerosi programmi applicativi già realizzati o per studiarne dei nuovi. Problemi già risolti includono la contabilità generale, la gestione del magazzino, la contabilità semplificata, la prenotazione elettronica degli appuntamenti, il listino prezzi on line.

Perché l'assistenza di una ditta che opera in Italia è per forza la migliore. Una garanzia che solo una ditta italiana può offrire: la certezza di una buona e completa assistenza.

Qualunque sia il problema la risposta è una sola: General Processor. La General Processor è vicina, telefona (al mattino) allo 055 - 43.85.27



in media un Modello T al giorno. Tecnicamente, il Modello T rimane molto interessante per l'architettura evoluta rispetto ai tempi. Esso era costruito con una scheda madre che conteneva la CPU e sulla quale erano posti moduli specifici per le varie funzioni su due bus diversi, uno dedicato alla memoria ed alla funzione video, l'altro dedicato all'ingresso-uscita (I/O).

Un paio di anni dopo il Model T subisce un sostanzioso restyling offrendo una soluzione dal design particolarmente curato e una rivisitazione del progetto per mantenerne le prestazioni in linea con il tempo.

Pari innovazioni e soluzioni ingegnose nei progetti del Black Star della Penta System ma anche dei sistemi di Olivetti. Erano anni dove le soluzioni non si trovavano belle e pronte da importare da Taiwan, bisognava rompersi il cervello per mettere assieme esigenze di economicità e prestazioni discrete. Certo non ci si poteva affidare al clock del processore per spremere qualche punto di benchmark, bisognava lavorare a livello del circuito stampato e soprattutto del software di base.

Non possiamo rimanere insensibili alla nostalgia che ci coglie ripercorrendo queste tappe fondamentali della nostra storia e indifferenti difronte al rinvenimento di una testimonianza concreta di un sistema dimenticato in cantina. Oppure all'emozione speranzosa che ci assale al rinvenimento di un voluminoso manuale ad anelli che descrive con una minuzia di particolari in un linguaggio non sempre adeguato, certe innovazioni software legate allo sviluppo del firmware, dei sistemi operativi e degli stessi compilatori.

L'informatica con la "I" maiuscola era allora identificata con il software di base mentre il software applicativo era quasi snobbato e considerato un sotto-prodotto. Non possiamo non notare il totale cambiamento di prospettiva cui assistiamo oggi quando diamo per scontato che la stampante si installi con i suoi driver in maniera del tutto automatica e come parliamo dell'ultimo Office o Photoshop come i veri oggetti cui prestare at-

tenzione, mentre installiamo patch dei sistemi sempre più voluminose e il cui contenuto ci è del tutto ignoto.

Siamo partiti da lontano e non vediamo la fine della strada. Qualcuno di noi ha questo dono di riuscire a volgersi indietro ad ammirare quello che giace alle proprie spalle addormentato ma non morto. Altri, meno fortunati secondo me, guardano solo in avanti e scalpitano in attesa di release sempre più sofisticate che forse non aggiugeranno nulla di veramente importante, ma alle quali pare non riescano a rinunciare.

La storia dei calcolatori italiani è ricca di esperienze ed opportunità e se questo accadeva anche in altri paesi come la Francia, la Svezia, l'Inghilterra e la Germania, possiamo ben concludere che l'Italia un suo mattoncino l'ha ben cementato nell'edificio che oggi chiamiamo genericamente "informatica".

[Sn]



Il sistema Black Star della Penta System (1982); sopra l'unità centrale alla quale veniva collegata la tastiera e il monitor, sotto la macchina aperta. Il computer è tutto contenuto in una scheda unica montata sopra uno dei floppy.



Retro Code

Turbo Pascal 1.0

Il fascino della prima release di quello che sarà un ambiente di sviluppo che ha fatto crescere una generazione (e forse anche più di una) di programmatori.

```
Logged drive: C

Work file:
Main file:

Edit     Compile  Run   Save

eXecute  Dir      Quit  compiler Options

Text:     0 bytes
Free: 62932 bytes

>_
```

L'IDE (chiamiamolo così) del Turbo Pascal 1.0

Sulle pagine di JN c'è stata più di una occasione per ospitare il Pascal e la versione della Borland in particolare, denominata fin dalla sua nascita con l'appellativo di "turbo". E in effetti proprio il turbo metteva questa implementazione del famoso linguaggio inventato da Niklaus E. Wirth, rispetto alle prestazioni da lumaca dei compilatori che si erano visti fino ad allora sui personal computer. Ad esempio l'UCSD Pascal nelle sue varie implementazioni e in particolare in quella Apple, non si può certo dire fosse una scheggia! Parimenti le implementazioni viste su CP/M, che dovevano fare i conti con un sistema operativo flessibile ma non certo orientato alle prestazioni.

Abbiamo parlato però finora solo della versione 7.0, chiamata "Borland Pascal", in pratica l'ultima ad uscire prima che la Borland cambiasse ambiente di sviluppo (Windows) e nome (Delphi) per il suo prodotto più famoso.

Vogliamo in questa occasione parlarvi della versione 1.0 del prodotto Borland, uscito per il sistema DOS sui PC della IBM e cloni.

Installazione

Si tratta di un ambiente di programmazione IDE, con tutti i distinguo del caso: niente finestre multiple e ricchi menù di funzioni. Appena pochi comandi che permettono di editare il sorgente, salvarlo, compilarlo ed infine eseguirlo.

```
TURBO Pascal installation menu.
Choose installation item from the following:

[S]creen installation  |  [C]ommand installation  |  e[X]it

Enter S, C, or X:
```

La schermata di installazione, serve per settare i parametri di funzionamento.

Già dalla distribuzione il programma rivela la sua frugalità: appena 150 Kb in tutto, compresi un paio di esempi.

Gli eseguibili sono solo tre: Turbo.com (che si porta dietro un overlay TURBOMSG.OVR) che è il programma vero e proprio; TINST.COM che è un menù-driven per associare i tasti alle funzioni dell'ambiente e settare alcuni parametri del compilatore, oltre che scegliere il tipo di monitor disponibile (80 o 40 colonne, bianco/nero o a colori); infine TLIST.COM che serve per listare un sorgente in modo formattato, soprattutto per la stampa su carta.

L'installazione avviene copiando il contenuto del floppy originale (da 5,25 pollici, 360 Kb di capacità) in una directory, ad esempio C:\TP.

Successivamente si lancia l'installatore per definire l'ambiente e il gioco è fatto!

Uso

Il lancio del programma TURBO.COM da inizio alla sessione di lavoro. La schermata iniziale è visibile nello screen di apertura nella pagina precedente. Come vedete le funzioni sono limitate, giusto il necessario per lavorare, senza fronzoli di sorta.

L'idea di base è la seguente: il programmatore interagisce con un "workspace" nel quale crea un nuovo sorgente o ne carica uno da disco. Tutto si svolge in memoria (il segreto della velocità di TP): editing, compilazione ed esecuzione sono le tre attività integrate che vengono eseguite "in punta di tastiera".

Un consiglio che ci sentiamo di dare a coloro che vogliono provare l'ebbrezza del vecchio fascino è quella di segnarsi le combinazioni di tasti fondamentali, come l'uscita dall'editor (Ctrl-K Ctrl-D): se ve lo scordate non riuscirete ad abbandonare l'operazione di editing se non spegnendo il PC!

L'editor, la componente con la quale si è costretti a convivere per la maggior parte del tempo, implementa la visione di Borland rispetto agli editor di testo, ma si

```
Choose one of the following displays

 0) Default display mode
 1) Monochrome display
 2) Color display 80x25
 3) Color display 40x25
 4) b/w   display 80x25
 5) b/w   display 40x25

Which display? (Enter no. or ^X to exit):
```

```
     Line 25    Col 1    Insert     Indent   C:CALC.PAS
{ This program is hereby donated to the public domain            }
{ for non commercial use only                                    }

{Here is a note to the compiler:                                 }
{$R-,U-,V-,X-,C-}

program MicroCalc;

const
  FXMax: Char  = 'G';  { Maximum number of columns in spread sheet   }
  FYMax        = 21;   { Maximum number of lines in spread sheet      }

type
  Anystring   = string[70];
  SheetIndex  = 'A'..'G';
  Attributes  = (Constant,Formula,Txt,OverWritten,Locked,Calculated);

{ The spreadsheet is made out of Cells every Cell is defined as      }
{ the following record:}

  CellRec     = record
    CellStatus: set of Attributes; { Status of cell (see type def.)  }
    Contents:   String[70];        { Contains a formula or some text }
    Value:      Real;              { Last calculated cell value      }
```

Siamo nell'editor del sorgente. Nemmeno un menù a disposizione: bisogna ricordarsi i tasti di controllo, altrimenti non se ne esce!

sente ancora l'influenza del CP/M e della gestione via terminale delle macchine. Ad esempio il movimento del cursore non conosce i tasti freccia (almeno nel default), mentre usa per questa funzionalità le combinazioni CRTL-S, CTRL-D, CTRL-E e CTRL-X individuando una "croce" fittizia sulla tastiera. Da notare l'uso di tasti "vicini" al Control, per non rendere troppo difficile l'abitudine ai programmatori.

L'unico aiuto viene dalla linea di stato, curiosamente posizionata in alto (siamo ormai abituati a vederle in basso questi flag di stato). In essa sono riportate le posizioni riga-colonna del cursore, il numero di linea, lo stato di alcuni flag (indentazione automatica e modalità inserimento rispetto alla sovrapposizione, oltre che il nome del sorgente.

Lasciato il sorgente basta il tasto "C" per compilare e "R" per eseguire il frutto del nostro lavoro. Il compilatore crea file di tipo COM, quindi non rilocabili, uno schema di organizzazione della memoria limitata (solo 64Kb per il codice e altrettanto per le variabili) se vogliamo, ma adatta all'epoca dei primi PC.

L'esempio contenuto sul floppy originale è un mini-spreadsheet chiamato mini-calc (vedere la figura della pagina a fronte) per abbiamo provato a compilare ed eseguire, senza trovare difficoltà alcuna.

Ovviamente non si tratta di un Excel, le possibilità sono limitate, del resto mica l'Excel si accontenta di 64 Kb di RAM!

L'esplorazione del sorgente di esempio, fra l'altro stupendamente commentato) fa intuire che l'interpretazione di Borland del Pascal fa uso massiccio di direttive di compilazione e si concede più di qualche licenza poetica per adeguare un linguaggio nato a scopi educational ad un ambiente veramente produttivo. Una lettura attenta ed esaustiva del manuale rivelerebbe sicuramente notevoli possibilità di realizzare applicazioni anche mediamente complesse con tanto di overlay, gestione errori, etc...

Conclusione

Diciamo che riusare questo storico software è decisamente educativo. Si ottiene una conferma dei ricordi che avevamo, cioè di un oggetto incredibilmente agile e veloce nell'uso e nell'esecuzione. All'epoca dell'uscita fece molto scalpore questo prodotto, proprio per l'innovativo approccio alla compilazione one-step e memory-oriented. Sappiamo oggi, con il senno di poi, che la fama era strameritata.

[Tn]

Il mini-calc in azione.

```
 A          B          C          D          E          F         G
 1    123.00
 2     12.00
 3     27.00
 4
 5=A1+A2+A3
 6
 7
 8
 9
10
11
12
13
14
15
16
17
18
19
20
21
A 5 Text                        Type / for Commands      AutoCalc: ON
```

Biblioteca

Dagli scaffali della nostra biblioteca, polverosi volumi ci parlano dell'informatica di una volta.

Pascal - manuale e standard del linguaggio



Chi per avventura della vita si sia trovato in mezzo alla rivoluzione informatica degli anni '80 con la necessità o la volontà di dover imparare un linguaggio di programmazione, non ha potuto certo fare a meno di imbattersi nel Pascal. Nel momento della sua nascita, attorno ai primi anni '70, l'autore di questo linguaggio, tale Niklaus Wirth, professore universitario a Zurigo, stava concretizzando le idee che da qualche anno riteneva valide per un insegnamento ottimale della difficile arte della programmazione.

Del Pascal sopravvivono oggi due filoni: il famosissimo Delphi della Borland e il meno noto al grande pubblico Object Pascal, una interpretazione in chiave moderna usata ad esempio nei tool del sistema operativo MAC OS X della Apple.

Imparare il Pascal, ricordo, è stato per il sottoscritto, come credo per molti altri, un rito di passaggio: si passava dal Basic, giocoforza imparato in quanto unico idioma dato a corredo delle macchine home dell'epoca, ad un linguaggio vero con compilatore e tutte le cose a posto.

Che emozione il lancio del UCSD sull'Apple II! E quante ore nel laboratorio di calcolo della facoltà di matematica dove mi intrufolavo spesso e volentieri (io studiavo fisica, ma solo i matematici potevano permettersi i lussuosi Apple, i fisici dovevano smanettare sul PDP-11 e poi sul Vax o su qualche workstation HP, certo più potenti, ma volete mettere il fascino del calcolatore personale?)

Così fui praticamente obbligato dal docente "titolare" del laboratorio, ad iscrivermi a TAMC (Teoria e Applicazione delle Macchine Calcolatrici), pena l'esclusione dalla sala "giochi" per gli anni a venire.

Il Pascal, con quel suo vezzo di scrivere l'assegnazione con la combinazione ":=" piuttosto che con il semplice "=", è stato un gran-

de linguaggio di programmazione; derivato da altri linguaggi come ALGOR e PL/1, ha lasciato dietro di se, oltre alle due implementazioni precedentemente citate, anche ADA, la cui poca fortuna è un'altro esempio di come le cose buone in informatica sono spesso destinate al dimenticatoio.

Quali le cose buone, oltre alla sintassi perfetta per l'insegnamento? Prima di tutto la pre-compilazione: Pascal è normalmente un linguaggio pre-compilato in una sorta di codice intermedio chiamato P-code. Questa sigla, che evidentemente sta per Pascal-code, viene usata tutt'oggi per indicare l'utilizzo di questa idea tecnologica. Java che è attualmente sulla cresta dell'onda, ha fatto propri i risultati del Pascal e li ha portati al suo massimo sviluppo; ebbene, su alcuni testi dedicati a Java si trova ancora il termine P-code ad indicare il codice intermedio. Il fatto di avere una pre-compilazione ha permesso lo sviluppo di compilatori per le piattaforme in uso negli anni '80/90.Altra cosa sicuramente buona e giusta è la codifica della sintassi molto rigorosa che giunge fino alla definizione dei codici di errore, opportumente progettati per lasciare spazio alle implementazioni proprietarie. Quest'ultima idea è stata ereditata dal COBOL, altro "mattone" fondamentale dell'informatica.

Di meno buono c'è una certa prolissità dei sorgenti, avulsa dalle moderne pratiche informatiche, e probabilmente una certa rigidità delle procedure di I/O.

La monografia, edita dal Gruppo Editoriale Jackson, è l'edizione italiana del manuale ufficiale, scritto in collaborazione con lo stesso Wirth.

Si tratta di una pubblicazione abbastanza anomala per il mercato italiano, dove sono sempre andate per la maggiore le pubblicazioni in stile "ti insegno passo passo come si fa...". Qui non c'è nulla di simile: una presentazione rigorosa, oserei dire "accademica" della sintassi del linguaggio, compresa l'assurda sintassi Backus-Naur (chi non sa di cosa sto parlando non ha perso nulla...), sciorinata definizione dopo definizione. Non manca, come si diceva prima, la puntuale esposizione degli errori e una stringata ma essenziale appendice F con esempi di programmazione classici quali il calcolo del minimo comun divisore, l'ultra-classico Sieve of Erathostene e la torre di Hanoi.

Un classico che non puo' mancare nella libreria di un appassionato di linguaggi di programmazione.

[Sn]

Scheda

Titolo:
Pascal: manuale e standard del linguaggio

Autore:
Kathleen Jensen
Nikoaus Wirth

Editore:
Gruppo Editoriale Jackson

Anno :
1985

Lingua:
Italiano

Pagine:
60

Soggetto:
Linguaggi di programmazione per calcolatori; Sintassi e formalismi.

Retro Linguaggi



La storia dell'informatica è stata anche la storia dei linguaggi di programmazione.

ABAP (parte 2)

Le variabili

Abbiamo visto la scorsa puntata come si fanno i primi passi nell'ambiente di sviluppo, l'ABAP Workbench.

La realizzazione di un programma di tipo "Report", cioè di semplice output, avviene in semplici passi ormai comuni a molti ambienti di programmazione integrata (i cosidetti IDE): scrittura del codice, controllo della sintassi, compilazione ed esecuzione. Nel caso dell'ABAP, essendo un semi-interpretato, non c'è una compilazione del codice ma essa avviene al primo richiamo del programma con una tecnica che viene definita "Just in Time", cioè qualdo serve.

E' necessario a questo punto arricchire il nostro bagaglio di conoscenza delle istruzioni elementari e dell'uso delle variabili, base di qualsiasi elaborazione.

Il primo programma che realizzeremo è il classico "Hello World!".

Il programma, listato nel box 1 introduce alcuni concetti base che incontriamo per la prima volta e che andiamo ad esaminare uno ad uno. Mi scuso con coloro che giudicheranno troppo elementari questi concetti e che preferiscono arrivare subito al sodo, ma la mia esperienza di insegnamento nell'ambito della programmazione mi conforta nel non dare per scontato nulla. Solo in questo modo infatti l'apprendimento si rivela completo e non, come purtroppo ho modo di notare spesso, lacunoso e pressapochista.

L'istruzione DATA definisce le variabili che seguono le usuali regole per i nomi nei linguaggi di programmazione: devono cominciare con una lettera, con possono contenere alcuni simboli speciali (-+/*?...). Il carattere "trattino" ("-") nei nomi di variabili serve come qualificatore nelle variabili strutturate che incontreremo più avanti ma che diamo un esempio per far capire meglio di cosa si tratta:

INDIRIZZO-NOME.

INDIRIZZO-VIA.

Sono due variabili (NOME e VIA)

Box 1.
Il programma "Hello World" in ABAP.

```
REPORT  z00_p_001.

DATA testo(20) TYPE c.

START-OF-SELECTION.

  MOVE 'HELLO WORLD OF ABAP!' TO testo.
  WRITE testo.

  testo = 'WELCOME TO SAP'.
  WRITE testo.
```

appartenenti ad una costruzione di dato strutturato (un RECORD) identificato con il nome INDIRIZZO.

Tornando alla variabile dichiarata con "DATA", essa ha nome "TESTO" (ABAP non è case-sensitive, cioè non distingue maiuscole da minuscole) è di tipo "stringa di caratteri" (type C significa tipo carattere) ed ha dimensione 20 caratteri. Significa che potrà contenere una sequenza di caratteri non superiore alla sua lunghezza.

Nel corpo del programma alla variabile sarà assegnato un valore, cioè una stringa di caratteri, che sarà stampata a video con l'istruzione WRITE.

Da notare che l'assegnazione del valore ad una variabile può avvenire sostanzialmente in due modi: il modo classicco ABAP, derivato dal COBOL, che consiste nell'istruzione MOVE <valore> TO <variabile> e quello più "moderno" ormai in uso praticamente in ogni linguaggio di programmazione

<variabile> = <valore>

Le due forme sono del tutto equivalenti anche se personalmente tendo ad usare la prima sintassi e non tanto per una questione di nostalgia, quanto per una chiarezza espositiva del codice stesso.

Le variabili si possono dichiarare in qualsiasi punto del codice, anche immediatamente prima dell'uso. In ogni caso esse sono "globali", cioè visibili all'intero programma. Per ragioni di chiarezza e ordine si usa dichiararle tutte all'inizio e così noi faremo nel prosieguo del corso.

Ci rimane da spiegare il significato della riga: "`START-OF-SELECTION.`*"*

Questa non è una istruzione ma una direttiva e serve ad individuare l'entry-point del programma. Come nel C il codice inizia con la funzione main(), così in ABAP c'è questa "label" che indica l'inizio del flusso principale.

Come in qualsiasi linguaggio di programmazione tipizzato, anche in ABAP esistono i dati "buil-in" o "elementari". Essi sono elencati nel box numero 2.

L'occupazione di memoria e la precisione sono standard su qualsiasi tipi di hardware/sistema operativo. Questo garantisce la trasportabilità del codice.

```
C      Carattere
I      Integer
F      Real Floating Point
P      Packed (numerico)
N      Carattere numerico
D      Data
T      Time
X      Esadecimale

STRING Stringa di caratteri a dimensione libera
```

Box 2.
I tipi di dato elementare riconosciuti dall'ABAP.

Stringhe

Il tipo di dato "STRING" non è in realtà un tipo elementare, fa parte delle strutture di appoggio inserite nel linguaggio, ma è giustamente di uso comune e lo abbiamo presentato qui proprio per questo motivo. Un tipo STRING rappresenta una sequenza di caratteri di lunghezza variabile (fino a 1024 caratteri).

Le due istruzioni DATA che seguono dichiarano la stessa variabile con nome STRINGA lunga 20 caratteri. Se il tipo di dato non è indicato il sistema assume il tipo carattere.

```
DATA STRINGA(20) TYPE C.
DATA STRINGA(20).
```

Nota: il compilatore segnalerà errore se si cerca di definire più volte la stessa variabile.

Altri esempi di dichiarazione di variabili.

```
DATA NUMERO(12) TYPE N.
DATA CON_VIRGOLA TYPE F.
DATA CON_DECIMALI(12) TYPE N DECIMAL 2.
```

L'ultima dichiarazione prevede per la variabile "CON_DECIMALI" una lunghezza complessiva di dodici cifre, due delle quali riservate ai decimali.

Istruzione di assegnazione:

```
MOVE 'TESTO CONTENUTO' TO STRINGA.
```

carica nella variabile il valore rappresentato da una stringa costante racchiusa fra apice singolo.

Nota: non esiste il problema dell'Overflow: il runtime controlla le lunghezze e tronca se necessario:

```
MOVE 'TESTO MOLTO LUNGO CHE NON CI STA'' NELLA VARIABILE'
TO STRINGA.
```

In STRINGA verranno inseriti solo i primi 20 caratteri:

```
WRITE STRINGA.
```

Stampa 'TESTO MOLTO LUNGO CH', cioè appunto i primi 20 caratteri.

E' possibile assegnare il valore iniziale al momento della dichiarazione:

```
DATA CAP(5) TYPE C VALUE '38100'.
DATA CITTA(20) VALUE 'TRENTO'.
```

Nel programma di esempio visibile nel Box 3 si utilizzano due variabili per contenere due testi che poi vengono stampati con un'unica istruzione WRITE.

Box 3.

```
REPORT Z00_P_002.

DATA TESTO(20) TYPE C.
DATA TESTO2 TYPE STRING.

START-OF-SELECTION.

  MOVE 'HELLO WORLD OF ABAP!' TO TESTO.
  MOVE 'WELCOME TO SAP' TO TESTO2.

  WRITE: TESTO, TESTO2.
```

Una delle caratteristiche della sintassi ABAP è che qualunque istruzione del linguaggio può essere applicata a più argomenti. In questo caso si fa uso del simbolo ':' (due punti) e gli argomenti vengono separati tramite una virgola.

Le variabili numeriche

Il tipo elementare 'N' dichiara una stringa che può contenere solo cifre (ad esempio il CAP, la partita IVA, un numero di telefono,...).

Dati numerici "binari" si dichiarano con i tipi I = Integer e F = floating point:

```
DATA INDICE TYPE I.
DATA RISULTATO TYPE F.
```

Dati Packed

Un tipo elementare numerico interessante e che crediamo sia sconosciuto ai più (almeno a coloro che non praticano il COBOL) è il tipo "Packed".

Serve per i calcoli "business"; possono rappresentare numeri, anche con decimali, molto grandi massimo 16 byte = 31 cifre (un semibyte è usato per il segno).

```
DATA BIG_NUMBER TYPE P
     DECIMAL 3.
MOVE '1234567890,12' TO P.
```

Tipi "calendario"

Per lavorare con i valori delle date del calendario, il tipo elementare 'D' viene rappresentato in memoria nella forma tradizionale "business", cioè come sequenza di caratteri numerici: YYYYMMGG (anno, mese, giorno). Questa rappresentazione è comoda per il confronto fra le date ed è pertanto usatissima in certi linguaggi. Infatti un semplice confronto di stringa indica immediatamente la mutua grandezza fra due date così rappresentate.

```
DATA BORN_DAY TYPE D
     VALUE '19970125'.
```

Rappresenta il 25 Gennaio dell'anno 1997.

Analogamente il tempo ha un suo tipo elementare: "T".

```
DATA ORA_DI_PARTENZA TYPE T
     VALUE '153247'.
```

I dati di tipo Time sono rappresentati nella forma 'HHMMSS'.

Valori esadecimali

```
DATA HEXCODE(3)  TYPE   X
     VALUE   'F9AA45'.

DATA STRINGAEX TYPE XSTRING.
```

La seconda dichiarazione è una variabile di tipo "byte stream", una sequenza di byte di lunghezza variabile (max 64 Kb).

```
REPORT  Z00_P_003                          .

* --- DEFINIZIONE DI DATI ELEMENTARI

DATA STRINGA1(20) TYPE C.
DATA STRINGA2 LIKE STRINGA1.
DATA CARATTERE VALUE 'X'.

START-OF-SELECTION.
* ----- ISTRUZIONI DI ASSEGNAZIONE
MOVE 'PRIMA STRINGA' TO STRINGA1.
MOVE 'SECONDA STRINGA' TO STRINGA2.

WRITE: STRINGA1, STRINGA2, CARATTERE.

* --- DEFINIZIONE DI DATI NUMERICI
DATA NUMERICO(5) TYPE N.
DATA N2 TYPE I.

MOVE '12345' TO NUMERICO.

WRITE: /, NUMERICO.

MOVE 12 TO N2.
WRITE: /, N2.
WRITE: N2, N2, N2, N2.

* --- CALCOLI
DATA N3 TYPE P DECIMALS 2.
DATA RESULT TYPE F.

MOVE '12.99' TO N3.
COMPUTE RESULT = N3 - N2.
WRITE: /, N3, ' - ', N2, ' = ', RESULT.
```

Box 4.

A fianco: l'output ottenuto.

```
Esempio dichiarazione variabili

16.02.2006                                  Esempio dichiarazione variabili
PRIMA STRINGA        SECONDA STRINGA      X
12345
          12          12          12          12          12
              12,99   -          12   =   9,9000000000000021E-01
```

Nel Box 4 abbiamo riportato un esempio di un programma completo che utilizza vari tipi di dato e li manipola stampandoli poi in output. Il risultato è visible nella figura sotto il listato.

Nota: l'istruzione "WRITE" produce l'output a video con una formattazione auto-definita che dipende dall'occupazione di spazio di ogni variabile. Questo comportamento può essere modificato inserendo delle direttive di formattazione nell'istruzione, direttive che esploreremo in dettaglio nel seguito. Per ora si noti che il carattere "/" (barra diritta o slash) determina un "a capo" della riga di stampa.

Si noti l'istruzione COMPUTE che serve per effettuare dei calcoli. In realtè questa sintassi p un refuso proveniente dal COBOL, così come le istruzioni ADD e SUBTRACT.

Più "modernamente" si può utilizzare la sintassi "di assegnazione del valore":

```
RESULT = N3 - N2.
```

Pointer

Il nome di una variabile può essere usato come puntatore alla memoria allocata per contenente il valore della variabile stessa.

Vediamo un esempio:

```
DATA DATA_DI_OGGI TYPE D
      VALUE '20050224'.

WRITE DATA_DI_OGGI.
WRITE DATA_DI_OGGI(4).
WRITE DATA_DI_OGGI+2(2).
```

La variabile si chiama "DATA_DI_OGGI" e viene caricata con un certo valore. Seguono tre righe di spampa nelle quali la variabile è utilizzata in modo differente.

La prima istruzione è una classica Write che stampa a video qualcosa come:

"24/02/2005"

La seconda dice di stampare solo i primi quattro caratteri della variabile:

"2005"

Ricordiamo con in memoria la data è conservata con una sequenza di otto byte nelle forma anno-mese-giorno.

Infine nella terza istruzione la sequenza "+2" che segue immediatamente il nome della variabile, sta ad indicare che si vuole partire con la stampa con un offset di due byte rispetto al puntatore della variabile in memoria. La stampa è "05".

Questa sintassi per l'aritmetica dei puntatori è un punto a sfavore dell'ABAP. Infatti è molto facile dimenticarsi degli spazi quando si scrive una formula di calcolo!

A = B + 2.

è diverso da:

A = B+2.

Anche questo secondo appuntamento con l'ambiente di programmazione SAP è giunto al termine. Questi primi approcci sono per forza di cose un po' tediosi, ma è essenziale entrare un poco per volta nelle logiche di questo linguaggio che ci riserverà delle vere e proprie sorprese, vedrete!

[Mx]

L'intervista

Vari personaggi e amici incontrati qua e la per una chiacchierata sul mondo del retro computing.

Intervista a Francesco A.

Abbiamo incontrato Francesco nel retro del suo negozio di Informatica situato in periferia di una cittadina di provincia non molto grande. Così mentre lui assemblava un PC per un cliente, una sera dopo l'ora di chiusura, lo abbiamo lasciato parlare a ruota libera, intervenendo qua e la per chiedere chiarimenti e riportare il discorso sul tema che più ci interessava: il retrocomputer.

JN.
Che tipo di computer retrò collezioni e quanti ne hai?

FA.
Esclusivamente home fino al 1990. Trovo che sia in queste macchine l'essenza dell'informatica o almeno della sua storia. Non so nemmeno io quanti ne ho di preciso, anche perché di molti modelli ho varie versioni... tu chiedi, e io ti dico se ce l'ho.

JN.
Immagino che quelli più comuni non valga la pena chiedere... l'Aquarius?

FA.
Ce l'ho...

JN.
...Compucolor...

FA.
Sì...

JN.
il DAI computer?

FA.
...Comprato su eBay da uno svedese...

JN.
Texas TI 99/4?

FA.
Facile.

JN.
Z80 Nuova Elettronica?

FA.
Questo mi manca, ma lo considero più una scheda a microprocessore che un home vero e proprio, non ci sono giochi ad esempio. Come non ho altre schede "famose", il nano-computer, l'SGS, l'Amico 2000, il Rockwell AIM65,...

JN.
In definitiva quanti fra marche e

modelli significativamente diversi?

FA.
...circa un centinaio, credo, più o meno.

JN.
Raccontaci di più, come hai iniziato, per esempio.

FA.
Ho cominciato anch'io verso i 15 anni con il Commodore 64 e sono poi approdato al PC quando i giochi per PC hanno soppiantato le altre piattaforme. Il mio interesse era puramente ludico, lo confesso. Mai avuta la pazienza di programmare; per me il computer è un oggetto tecnologico come tanti altri, che so il telefonino, la televisione. Chiaro che per il lavoro che faccio devo smanettare con i sistemi operativi, qualche volta con gli script, ma poco altro.

JN.
Pensavo che uno che mette su un negozio di informatica è per forza di cose un appassionato "a tutto tondo", per dire così!

FA.
Bhe, siamo in due: il mio socio fa la parte più sistemistica e supporta le installazioni e qualche limitato sviluppo. Io sono soprattutto un hardwarista.

JN.
Quindi giochi a volontà, immagino...

FA.
Credo di possedere una collezione di giochi, parliamo di originali, che forse hanno pochi in Italia. La cosa era cominciata con il 64, poi le varie console Playstation, Sega, Dremcast... le ho avute tutte, credo. Poi con il PC è stata un'esplosione, anche grazie al fatto che a 18 anni, appena uscito dalle superiori ho trovato lavoro in un negozio di elettrodomestici abbastanza grande. Io mi occupavo della parte PC-Console e giochi, appunto. Quando il gioco non vendeva più, veniva eliminato dagli scaffali e buttato nella spazzatura. Io recuperavo... Così mi sono trovato un incredibile quantità di titoli quando lasciai il lavoro per mettermi in proprio assieme al mio socio.

JN.
Quindi quando recuperi un vecchio home cerchi di procurarti i giochi, immagino.

FA.
Proprio così. Per alcune piattaforme, come ad esempio lo Spectrum e lo stesso 64, la cosa è scoraggiante, nel senso che ce ne sono talmente tanti! Impossibile fare una raccolta completa, ma ci sto provando.

JN.

E come ti procuri macchine e software? Quali sono i tuoi canali di approvvigionamento?

FA.

Sono in contatto con un gruppo di persone via Internet, conosciute a varie fiere, Marzaglia per esempio, che ho frequentato in passato. Ormai il canale migliore è sicuramente eBay, anche perché non ho più così tanto tempo per "andare a caccia" e sulla rete si può trovare di tutto anche a prezzi onesti. Gente con la quale ho fatto "affari" in passato, capita che mi mandino una mail per offrirmi i sistemi o il software prima di metterlo su eBay.

A differenza della maggior parte delle persone che frequentano il mondo del retrocomputer, almeno a quanto leggo sui forum e nel gruppo it.comp.retrocomputer, io non tiro sul prezzo più di tanto. Non sono ipocrita, se una macchina mi manca ed è in buono stato caccio volentieri un centinaio di euro o anche di più per aggiudicarmela.

JN.

Sfido che i venditori te le offrano prima...

FA.

Lo sò, adesso mi dirai che rovino la piazza, che pagando troppo i vecchi sistemi creo l'illusione che valgano veramente centinaia di Euro. Sai cosa ti dico: non me ne importa nulla! Il valore per me non è dell'oggetto reale, il valore è puramente venale, da collezionista. Se qualcuno mi offre quel particolare modello che mi manca non so resistere... Il problema è che ormai non è che me ne offrano più tanti... in giro sembra siano rimaste le cose più note e basta. Se vuoi una valanga di Commodore 64 te ne posso procurare un centinaio nel giro di una settimana.

JN.

No grazie, ho già dato :-)

FA.

Non so come la pensi tu, ma io credo che non ci sia più nel gruppo dei retrocomputeristi quella sorta di spirito di appartenenza che percepivo una volta. Ad esempio sul newgroup icr non ti puoi più fidare a mettere su un tuo parere che subito in dieci ti attaccano, non per controbattere con argomentazioni, ma mi sembra più che altro per sputtanarti... il mondo è cambiato.

JN.

Non so che dire. Credo che se uno non frequenta con assiduità un gruppo, poi perda in poco tempo il senso di appartenenza che tu dicevi: magari vecchi interlocutori non scrivono più, mentre altri sembrano farla da padroni... difficile giudicare, rimane comunque l'unico canale nazionale dove si parla di retro computer.

FA.

Sarà. Forse, come dicevo, non ho più tanto tempo o forse invecchiando sono cambiato.

JN.

Mi incuriosisce il fatto che ti occupi sia di vecchie macchine che di quelle nuovissime che immagino ti commissionano i clienti.

FA.

Puoi giurarci, siamo specializzati in modding e in overclocking. Se vuoi un sistema affidabile con due schede video in parallelo con una grafica da urlo, devi venire da me!

JN.

Grazie, no. Ci verrei se decidi di liberarti di qualche pezzo della tua collezione... per i nuovi sistemi io vado con Apple e ho eliminato il problema dell'affidabilità alla radice!

FA.

Già, ci abbiamo provato a vendere Apple, ma il ricarico è ridicolo. Pensa che vendendo a prezzo di listino un portatile dal costo di quasi 1400 Euro, il nostro guadagno era inferiore a 15 Euro! Impossibile a queste condizioni!

JN.

Tornando al retrò, partecipi a mostre o eventi dedicati alla retro-informatica?

FA.

Qualche volta mi è stato chiesto, ma l'industria del computer non è così vecchia da suscitare meraviglia per i sistemi più vecchi, almeno non nel grande pubblico. Anni fa si era ipotizzato con un paio di amici di fare un museo, ma ci siamo arenati subito nel trovare il posto adatto. Ci vuole uno spazio notevole e il Comune prima ci ha illusi, ma poi ha pensato bene di rimangiarsi le promesse. Tutto sommato ora penso sia stato meglio così: ti immagini il tempo per gestire uno spazio espositivo?

JN.

Pensi che questa possa comunque essere una strada possibile per la tua collezione?

FA.

Sì e no. Da una parte è evidente che chiunque raccoglie oggetti tende a pensare che un giorno la sua raccolta sia talmente importante da meritare un posto espositivo permanente. Dall'altra la mia è una collezione che ha ragione di essere perché composta anche da software oltre che da hardware. Come si potrebbe esporre i giochi? Con qualche immagine, non certo dando la possibilità alle persone di giocare...

JN.

Ci sono gli emulatori, cosa ne pensi?

FA.

Non me ne intendo molto ad essere sincero. Vedo che le tecniche di emulazione sono molto sofisticate oggi e certo tecnicamente è possibile emulare tutto, però non so se sia la cosa più giusta da fare. In fondo anche i quadri si possono copiare, ma la gente vuole vedere l'originale.

JN.

Già, il paragone non fa una piega. Poi c'è il problema del copyrights!

FA.

Non ne parliamo! Io ho dovuto spostare la mia collezione da mia sorella perché una volta la finanza ci ha perquisito il negozio in cerca di software clonato, che noi non trattiamo ben inteso, ma ho paura che se mi arrivano in casa sia dura dimostrare che è tutta roba non commerciale!

JN.

Già, per chi ha una attività commerciale nel settore potrebbe essere sospetto avere una collezione come la tua...

Ti ringrazio a nome di tutti i lettori per averci dedicato del tempo e per averci permesso di pubblicare questa conversazione. Ti auguro ovviamente di continuare a ricavare soddisfazione nel tuo hobby.

FA.

Grazie, saluto tutti con affetto.

BBS

A colloquio con i lettori

L'indice generale 2006-2007

Dopo due anni dalla nascita di Jurassic News, a noi della redazione è parso giusto fare il punto del lavoro fatto e presentare possibili scenari futuri.

Quale modo migliore della pubblicazione dell'indice generale che trovate di seguito?

Rubrica	Titolo	Autore	Numero
Editoriale	Welcome to the past	Tn	1
	Di qualche accadimento	Tn	2
	Terzo numero di JN	Tn	3
	Go JN, go	Tn	4
	Give me five	Tn	5
	Solo chi ha nuove idee sopravvive	Tn	6
	Anno nuovo	Tn	7
	To be free	Tn	8
	Novità in vista	Tn	9
	Abbondanza e carestia	Tn	10
	La svolta	Tn	11
	Miti, eroi e soldati	Tn	12
	Orgoglio e umiltà	Tn	13
Totale	13		

Rubrica	Titolo	Autore	Numero
Edicola	Retro Gamer	L2	5
	La Gazzetta del Pirata	Sn	6
	Retro	Sn	7
	DEV.	Sn	8
	Pluto Journal	Sn	9
	Vintage Technology Magazine	Sn	11
	Magazine ZX	Sn	13
Totale	7		

Rubrica	Titolo	Autore	Numero
Retro Computing	Istruzioni per l'uso (I)	Tn	1
	Istruzioni per l'uso (I)	Tn	2
	Istruzioni per l'uso (I)	Tn	3
	Istruzioni per l'uso (I)	Tn	4
	Il mercato e i mercanti	Tn	5
	Calma, ce n'è per tutti	Tn	6
	Non di solo hardware	Tn	7
	Trashing, burning right	Tn	8
	Essere Liberi	Tn	9
	L'archivista	Tn	10
	L'etica e il restauro	Tn	11
	L'entusiasmo…	Tn	12
	…e la delusione	Tn	13
Totale	13		

Rubrica	Titolo	Autore	Numero
Il Racconto	Lory	Bs	3
	La megaditta, Genesi	Mm	4
	La grande impresa	Mm	5
	Affonda la flotta!	Mm	6
	Nella terra degli Unni	Mm	7
	Il mega prograamma	Mm	8
	Una giornata di Ivan Ivanovich	Bs	9
	Il Mega direttore galattico	Mm	10
	Mascalzone latino	Mm	12
Totale	9		

Rubrica	Titolo	Autore	Numero
Videoteca	Pirates of Silicon Valley	Sn	8
	2010 l'anno del contatto	Sn	9
	War Games	Sn	11
	Wings Commander	Sn	13
Totale	4		

Rubrica	Titolo	Autore	Numero
	MC microcomputer	Sn	1
	Re-Load Magazine	Sn	2
	Still Alive	Sn	2
	Bit	Sn	3
	Telematicus	Sn	4
	Amiga Magazine	Sn	5
Retro Riviste	Computer & Video Games	Sn	6
	Big K	Sn	7
	Papersoft	Sn	8
	Olivetti R&T Review	Sn	9
	Nibble Magazine	Sn	10
	Logica 2000	Sn	11
	m&p computer	Sn	12
	K	Sn	13
Totale	14		

Rubrica	Titolo	Autore	Numero
	Retro Gamer	L2	5
	La Gazzetta del Pirata	Sn	6
	Retro	Sn	7
Edicola	DEV.	Sn	8
	Pluto Journal	Sn	9
	Vintage Technology Magazine	Sn	11
	Magazine ZX	Sn	13
Totale	7		

Rubrica	Titolo	Autore	Numero
	Retro Gamer	L2	5
	La Gazzetta del Pirata	Sn	6
	Retro	Sn	7
Edicola	DEV.	Sn	8
	Pluto Journal	Sn	9
	Vintage Technology Magazine	Sn	11
	Magazine ZX	Sn	13
Totale	7		

Rubrica	Titolo	Autore	Numero
Retro Computing	Istruzioni per l'uso (I)	Tn	1
	Istruzioni per l'uso (I)	Tn	2
	Istruzioni per l'uso (I)	Tn	3
	Istruzioni per l'uso (I)	Tn	4
	Il mercato e i mercanti	Tn	5
	Calma, ce n'è per tutti	Tn	6
	Non di solo hardware	Tn	7
	Trashing, burning right	Tn	8
	Essere Liberi	Tn	9
	L'archivista	Tn	10
	L'etica e il restauro	Tn	11
	L'entusiasmo…	Tn	12
	…e la delusione	Tn	13
Totale	13		

Rubrica	Titolo	Autore	Numero
Il Racconto	Lory	Bs	3
	La megaditta, Genesi	Mm	4
	La grande impresa	Mm	5
	Affonda la flotta!	Mm	6
	Nella terra degli Unni	Mm	7
	Il mega prograamma	Mm	8
	Una giornata di Ivan Ivanovich	Bs	9
	Il Mega direttore galattico	Mm	10
	Mascalzone latino	Mm	12
Totale	9		

Rubrica	Titolo	Autore	Numero
Videoteca	Pirates of Silicon Valley	Sn	8
	2010 l'anno del contatto	Sn	9
	War Games	Sn	11
	Wings Commander	Sn	13
Totale	4		